


# ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ЕЕ ПРИМЕНЕНИЕ

Новое в жизни, науке, технике

Подписная научно-популярная серия

Издается ежемесячно с 1988 г.

## Семейство отечественных ЭВМ

1988

7

Новое
в жизни,
науке,
технике

Подписная
научно-
популярная
серия

7/1988

Издается
ежемесячно
с 1988 г.

# ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ЕЕ ПРИМЕНЕНИЕ

## СЕМЕЙСТВО ОТЕЧЕСТВЕННЫХ ЭВМ

## Микро-ЭВМ семейства БК

## В НОМЕРЕ

3 Ю. Я. КУЗЬМИН
ОБМЕН ОПЫТОМ
БЫТОВОЙ КОМПЬЮТЕР БК-0010 —
НАДЕЖДЫ И ВОЗМОЖНОСТИ

13 Н. ГЕРМАН
ВСЕ — ЗА КОМПЬЮТЕРЫ!

25 Б. Б. МАТВЕЕВ
СПАГЕТТИ В КОМПЬЮТЕРИЗАЦИИ

41 С. М. КОСЕНКОВ
НОВЫЕ МОДЕЛИ СЕМЕЙСТВА БК



Издательство
„Знание“
Москва
1988

ББК32.97
С30

# АВТОРЫ ВЫПУСКА

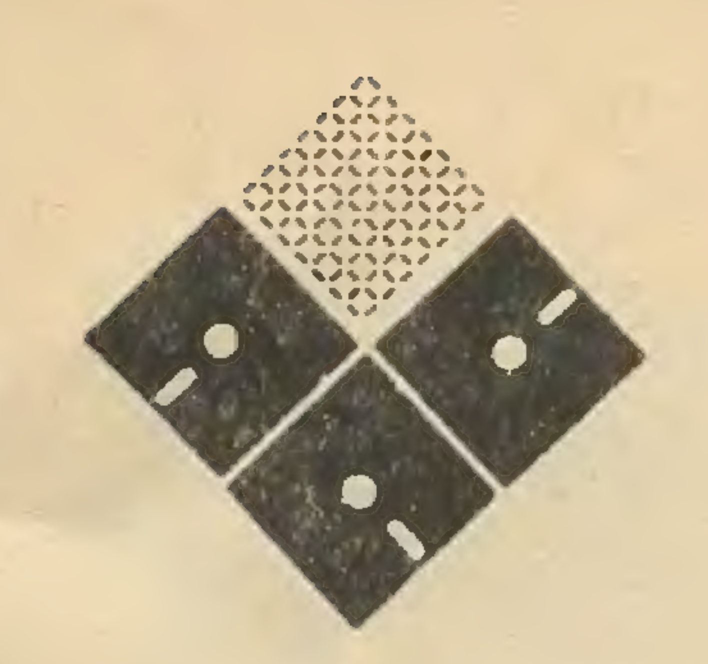
КУЗЬМИН Юрий Яковлевич — кандидат физико-математических наук, заведующий лабораторией НИИ физики твердого тела Латвийского государственного университета в г. Риге.

Н. ГЕРМАН — кандидат физико-математических наук, работает в Институте высоких температур АН СССР.

МАТВЕЕВ Борис Борисович — физик, работает в Институте ядерных исследований АН СССР.

КОСЕНКОВ Сергей Михайлович — главный технолог завода „Экситон".

**Редактор Б. М. ВАСИЛЬЕВ**



**Первый отечественный компьютер БК-0010 находится в розничной торговле. Появилась новая категория пользователей ЭВМ. Легкий и небольшой прибор, который вместе с блоком питания и портативным магнитофоном, можно положить в портфель и прибыть на лекции по вычислительной технике или на собрание членов клуба ЭВМ и через несколько минут начать работу со своей ЭВМ. Это впечатляет! БК-0010, как следует из названия (БК — бытовой компьютер), ориентирован на использование в быту, а значит, на бытовую технику. Действительно, чтобы записать разработанные программы, достаточно обычного магнитофона, а чтобы увидеть текст программ и результаты их работы, нужно иметь обычный телевизор. Конечно, при этом могут возникнуть некоторые проблемы. Опытом их решения и хотелось бы поделиться.**




Ю. Я. КУЗЬМИН

# БЫТОВОЙ КОМПЬЮТЕР БК-0010 — НАДЕЖДЫ И ВОЗМОЖНОСТИ

## Телевизор для БК-0010

Все начинается с телевизора. В отличие от калькуляторов, где индикатор встроен, БК-0010 использует индикатор на электронной трубке. Поскольку такая трубка имеется у любого телевизора, то остается лишь подключить его к БК-0010.

Если ваш телевизор имеет вход «видео», то нужно вначале установить режим работы телевизора на прием видеосигнала. Это делается либо извне, например подачей напряжения на реле переключателя режима, либо в самом телевизоре, переставив перемычку видеосигнала. Лучше всего обратиться с этой целью в телеателье, тем более что может понадобиться установить в телевизоре дополнительные элементы.

Если ваш телевизор не имеет входа «видео», то можно посоветовать один из нескольких вариантов подключения БК-0010. Проще всего купить либо изготовить преобразователь видеосигнала в телевизионный антенный сигнал. Такой преобразователь избавит вас от необходимости переделывать телевизор. Примитивный преобразователь делается на основе одного транзистора. Схему можно найти в радиолюбительской литературе по телеиграм. БК-0010 с таким преобразователем превращается в подобие домашнего телецентра, передающего вместо привычных изображений текстовую и графическую информацию. Учитывая это, будьте осторожны, чтобы БК-0010 не стал источником помех для телевизоров ваших соседей. Такое может случиться, если провод преобразователя соединится с коллективной антенной, а также если длина и расположение провода окажутся оптимальными для радиоизлучения.

Если у вас нет преобразователя видеосигнала, но есть паяльник, конденсатор 2-20 мкФ, кусочек кабеля и опыт работы со схемами, то вы можете сделать вход «видео» самостоятельно. Для этого нужно вскрыть телевизор, найти точку за видеодетектором и припаять к ней один конец конденсатора, а второй — к антенному гнезду, например, вместо гнезда 1:10 или к новому гнезду, установленному вами специально для БК-0010.

Получение цветных картинок из БК-0010 возможно только подключением к каналу цветности телевизора и только

из тех БК-0010, которые имеют RGB-выход. Некоторые телевизоры имеют RGB-выход, но большинство потребуют переделки. В транзисторных телевизорах нужно подключиться через конденсаторы 1-20 мкФ к цветовым каналам, соблюдая полярность сигналов RGB. Для ламповых телевизоров придется делать согласователи. В любом случае, принимая решение о цветном телевизоре для БК-0010, еще раз взвесьте все обстоятельства: стоимость телевизора, сложность переделки, трудности ремонта и, наконец, малое число «цветных» программ и сложность их составления. Учтите также, что тексты на цветном телевизоре видны много хуже, чем на черно-белом.

Какой бы вы ни выбрали телевизор — лучше, если он будет служить только для персонального компьютера.

После того как решены все проблемы с телевизором, наступает ответственный момент. Включите телевизор, убедитесь, что он работает, проверьте, нет ли на антенном (или «видео») входе напряжения, могущего повредить БК-0010 при подключении. Если все в порядке, то подключите компьютер к телевизору. Затем включите питание БК-0010. На экране телевизора должен появиться текст: «Готовность к работе». С этого момента можно приступить к работе. Если сообщения нет, то быстро выключите БК-0010 и еще раз тщательно все проверьте.

## Магнитофон для БК-0010

С магнитофоном все намного проще. Пригоден практически любой современный магнитофон. Рекомендую монофонические кассетные магнитофоны малого размера, имеющие счетчики расхода ленты и возможность отключения АРУ. Стереомагнитофоны дороже и более капризны в работе. Отсутствие счетчика ленты затрудняет поиск программ. Представьте, что программа записана в середине ленты и нужно «вслепую» найти ее начало. Без счетчика это отнимет много времени. Автоматическая регулировка усиления может мешать из-за «вытягивания» уровня сигнала там, где этого не требуется.

Как подготовить магнитофон для работы с БК-0010? Прежде всего надо убедиться, что он исправен и хорошо работает на обычных фонограммах — звук не плывет, нет тресков и других лишних шумов, ленту не заедает. Подключите такой магнитофон к БК-0010, установите режим записи и включите усилитель на полную мощность. Если все исправно, то послышится монотонное шипение.

Очень важное значение имеет правильная настройка головки чтения записи магнитофона. Точная ее установка производится по специальной записи на ленте. Такая лента имеется на заводах фирмы «Мелодия» и у изготовителей магнитофонов. В худшем случае можно воспользоваться качественной фирменной записью и отрегулировать головку так, чтобы хорошо слышались звонкие звуки. Заметьте это положение головки и возвращайтесь к нему после каждой вынужденной перестройки. Старайтесь, чтобы все ваши программы были записаны при одном положении головки магнитофона. Программа, записанная при других положениях, легко распознается по глухому звучанию и невозможности ввести ее в БК-0010 без смещения головки.

Большое значение имеет уровень выходного сигнала магнитофона. Он должен быть в пределах 0,6—0,8 В. Это достигается при записи с небольшой перегрузкой по уровню сигнала (АРУ отключается).

Не всегда причиной неудачного ввода программы в БК-0010 является сам магнитофон. Вот некоторые другие возможные причины.

1. Помехи от телевизора. Они могут поступать от схем телевизора по соединительному кабелю или вне его. Это проявляется как гудящий звук, возникающий чаще всего от цветного телевизора. Его можно уменьшить, либо удалив БК-0010 от телевизора, либо меняя угол их взаимного расположения.

2. Помехи от тиристорных схем. Они возникают, если в сеть, питающую БК-0010, включены приборы, управляемые

от мощных тиристорных переключателей. Чаще всего такое случается в заводских и лабораторных условиях.

3. Помехи от радиостанций, ретрансляторов и т. п. Уровень сигнала от них может быть настолько велик, что БК-0010 просто не сможет работать.

4. Плохие ленты или лентопротяжный механизм. Они могут иметь некачественный магнитный слой или заедающий механизм. Последнее случается чаще всего. Чтобы отбраковать ленту, поставьте ее в исправный магнитофон и прокрутите в обе стороны. Если лента перемотается бесшумно, то кассета исправна. Если будет слышен «царапающий» звук, то она чаще всего непригодна. Некоторые ленты при такой проверке даже останавливаются.

5. Если старая запись плохо стерта, то это также может быть причиной сбоев при чтении программы. Лучше всего предварительно стереть старую запись.

Как видим из беглого рассмотрения проблемы записи программ, вам потребуется более строгое отношение к технике, чем при обычной работе с фонограммами. Дополнительно еще несколько советов.

1. Во время записи программы всегда делайте дубль.

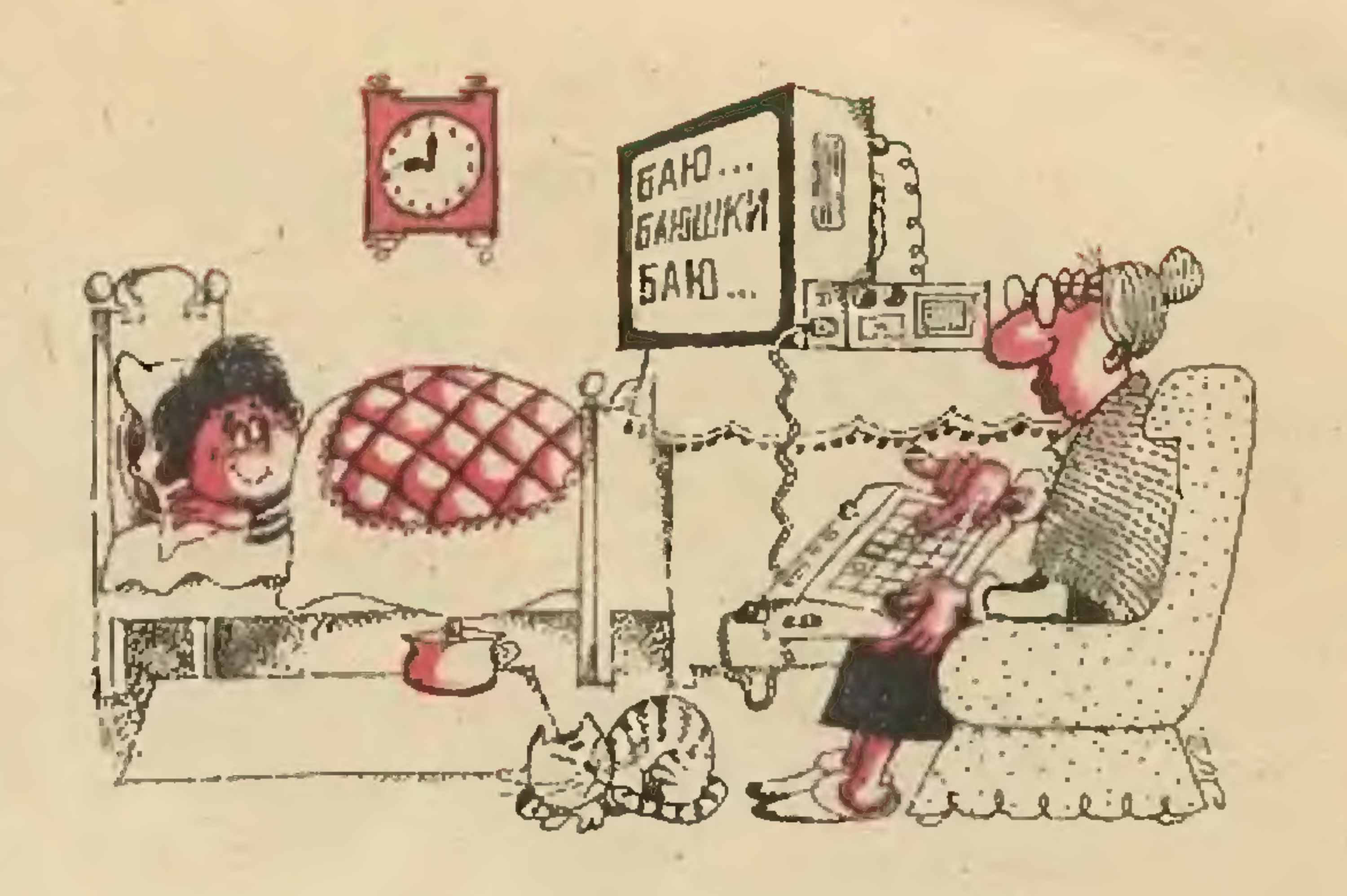
2. Всегда после записи сразу проверяйте ее, пока программа еще в памяти.

3. Производите промежуточную запись программ во время их ввода с пульта. Можно установить за правило записывать, например, при добавлении каждых 40 строк. Это избавит вас от повторения набора программы при неожиданных отказах.

4. Если запись не удается прочитать, попробуйте ввести ее с одного из каналов стереомагнитофона.

5. Если у вас нет кассетного магнитофона, пригоден и бобинный.

6. Если вы хотите сэкономить пленку и время считывания программы, то перепешите ее на пониженной скорости, а затем запишите на удвоенной. Получится двойное «сжатие» записи. БК-0010 обычно считывает такую запись.




## Параметры БК-0010

Возможности любой ЭВМ прежде всего определяются ее компонентами. Сердцем БК-0010 является микропроцессор К1801 (вычислительная схема). Именно он исполняет арифметические, логические, обменные и другие операции. В данном случае скорость их выполнения доходит до 300 000 операций в секунду. Марка микропроцессора та, что используется в профессиональных микро-ЭВМ типа ДВК. Система операций К1801 практически совпадает с еще более мощными мини-ЭВМ типа СМ-4 и СМ-1420. Это значит, что БК-0010 находится в одном ряду с современными ЭВМ и программы, составленные на БК-0010, можно с небольшими изменениями использовать на этих ЭВМ. Можно и, наоборот, использовать мощные ЭВМ для разработки программ для БК-0010. Это обычно делается на языке МАКРОАССЕМБЛЕР.

В чем же уступает БК-0010 этим ЭВМ? Прежде всего БК-0010 уступает профессиональным ЭВМ по объему оперативной памяти. Она здесь всего 16 тыс. байт (1 байт позволяет запомнить одну букву). Если, например, в такую память вводить только тексты, то войдет не более шести страниц книги из серии «Библиотека всемирной литературы». У БК-0010 есть, правда, режим, когда память можно расширить до 28 тыс. байт в ущерб выводимой на экран телевизора информации, но этот режим не всегда удобен. В оперативной памяти хранятся программы и данные, нужные ЭВМ для непосредственной работы. Поиск информации здесь происходит чрезвычайно быстро. Однако попадает сюда информация

вначале с пульта ЭВМ или из магнитофона, если она была туда предварительно записана.

Долговременная память на основе магнитофона медленная. Например, чтобы заполнить все 16 тыс. байт, нужно ждать считывания около 100 с. Зато на одной кассете можно записать сразу до нескольких сот страниц текста программ. Обычно для этих целей в профессиональных ЭВМ используют накопители на магнитных гибких дисках, которые позволяют находить любую информацию за доли секунды.

Итак, малый объем оперативной памяти и отсутствие дисков — важнейшие ограничения БК-0010 по сравнению с профессиональными микро-ЭВМ. Но в то же время БК-0010 имеет и преимущества. Главные из них — массовость, малые размеры, вес и стоимость. Кроме того, БК-0010 располагает возможностью выдачи графиков, рисунков и звуков. Еще совсем недавно эти возможности были у редких профессиональных ЭВМ, и они стоили десятки тысяч рублей.

Для ввода информации в БК-0010 используется клавиатура, совмещенная с корпусом. Она представляет собой набор контактных переключателей, покрытых плоской пленкой с нанесенными на нее символами. Чтобы ввести код символа в ЭВМ, нужно нажать на пленку с усилием, достаточным для переключения. Сразу отметим, что тут есть как отрицательные, так и положительные моменты. Во-первых, усилие иногда может быть значительным, особенно если пленка толстая и находится высоко над кнопками. Во-вторых, пленка быстро портится. В-третьих, при сильной электризации (например, трением) возможен пробой сквозь пленку микросхемы, обслуживающей клавиатуру. Наконец, профессионалам не нравится отсутствие свободного хода кнопок при их нажатии — «жесткость» кнопок БК-0010.

Несомненным достоинством такой клавиатуры является отсутствие щелей, в которые могут попасть мелкие предметы и испортить ЭВМ. Другой плюс — легкость замены символов пульта на новые простым наложением другой пленки (или полоски бумаги). Такой вариант был практически опробован и показал себя с самой лучшей стороны. Например, накладывались полоски с рисунками вместо букв (так можно учить ребенка азбуке). Можно также наложить набор символов национального алфавита (латышский, армянский, грузинский...) и работать с национальными текстами. Правда, для этого еще нужен и программируемый алфавит. К сожалению, в БК-0010 адрес алфавита постоянный, поэтому приходится использовать специальные программы для вывода новых символов. Пакет таких программ был создан и получил название системы «Рига-микро». В этот пакет дополнительно к символам БК-0010 (латинские, русские, цифры, пунктуация, спецграфика) можно вводить любой нужный алфавит. Все символы можно выводить в БК-0010 в виде 24 строк, каждая по 32 или 64 символа. В случае режима вывода 32 символов цвет каждого из них можно задавать отдельно. В этом же режиме выводятся цветные графики: 256 точек по горизонтали и 240 по вертикали. При режиме 64 символов эти числа составляют 512 и 24 соответственно.

Таким образом, БК-0010 позволяет программировать текстовую информацию с русскими и латинскими символами и включать в нее графическую информацию произвольной формы, что очень ценно для программ игрового, учебного и делового характера.

Интересной особенностью является возможность подключения БК-0010 к электронным устройствам. Для этой цели предусмотрены два плоских разъема. Один из них служит для подключения профессиональных устройств, предусматривающих канал обмена информацией ИРПС. Другой разъем — это 16 контактов, с которых можно принять сигналы в БК-0010, и 16 контактов, на которые можно выдать сигналы из БК-0010 по заложенной программе. Этот разъем назван портом. Через порт подключаются регуляторы световых приборов, контактные датчики, тепловые регуляторы и т. п. В научных целях этот порт используют для подключения

простых измерительных и управляющих приборов. Дома к нему можно подключить разные модели, например детскую железную дорогу. Большой интерес представит БК-0010 и для увлекающихся цветомузыкой.

Заканчивая рассмотрение технических особенностей БК-0010, укажем на возможность использования так называемых жестких программ. Для них предусмотрен специальный отсек. Ввод такой программы не требует магнитофона, нужно просто вставить микросхему с программой в гнездо отсека. Пока что такие микросхемы содержат трансляторы с языков ФОКАЛ и БЕЙСИК, но ожидается поступление и других программ (игр, отладчиков, редакторов и т. п.).

## Языки программирования БК-0010

В микросхеме постоянной памяти БК-0010 находится транслятор с одного из языков программирования. Это либо ФОКАЛ, либо более популярный в мире язык — БЕЙСИК.

К настоящему времени для БК-0010 сделано много программ на разных языках, в том числе и в машинных кодах. Если вам попалась такая программа, то вы должны уметь ее ввести и запустить на исполнение. Для этого нужно освоить процедуру перехода к специальному режиму ввода и пуска машинных программ. Процедура для ФОКАЛа и БЕЙСИКа различна. В первом случае вы должны нажать клавиши ЗАГЛ, ЛАТ, Р, пробел, М, ВВОД, во втором — ЗАГЛ, ЛАТ, М, О, *N*, *I*, Т, ВВОД. В ответ на выдачу ЭВМ вопросительного знака нажмите М. БК-0010 запросит «ИМЯ?», и тут нужно ввести имя программы на ленте, нажав ВВОД, запустить магнитофон на воспроизведение программы. Когда звук кончится (и программа тоже), вновь появится «?». В ответ нажмите РУС, С, ВВОД, если в начале программы находится первая команда. После этого программа начнет исполняться. Остановить программу можно, нажав клавишу СТОП.

Кроме трансляторов с языков ФОКАЛ и БЕЙСИК, в настоящее время доступны и другие. Все они вводятся с магнитной ленты указанным выше способом. Автору этой статьи известны трансляторы ФОРТ и АССЕМБЛЕР для БК-0010.

## Т-язык

Язык был придуман автором этой статьи в 1985 г. для БК-0010 специально для создания обучающих программ и игр, а также для документирования программ. Буква «Т» взята из слова «текст». Дело в том, что назначение Т-языка — создание программированных текстов, отличающихся от обычных включением диалога автора с читателем, динамических иллюстраций и звуковых эффектов. От программ эти тексты отличаются прежде всего тем, что основным элементом здесь является не оператор (приказ ЭВМ), а кадр — доза текста на любом естественном языке. Кадры выглядят так же, как обычные тексты в книге, в частности, они могут содержать и рисунки. Указания ЭВМ помещаются между кадрами в виде особых строк, которые начинаются восклицательным знаком, он как бы говорит ЭВМ: «Внимание! Это строка для тебя». За восклицательным знаком идут символ оператора и необходимые параметры. Например, согласно строке

```
!? , село, деревня
```

ЭВМ будет искать в ответе человека, только что введенном в ЭВМ, слово «село» или «деревня».

А согласно строке

```
!:+      верн
```

ЭВМ будет искать в программированном тексте строку, которая начинается сочетанием «верн», если результат поиска слов в ответе человека был положительным.

Транслятор с Т-языка и программа ввода текстов в ЭВМ названы системой «Рига-микро». Отличительной особенностью системы является возможность создания практически любых алфавитов, в том числе всех республик СССР. Это особенно важно для обеспечения

национальной письменности при компьютеризации образования и быта.

В настоящее время создано большое количество программированных текстов по самым различным темам, и они разошлись по многим нашим городам. Для тех, кто их имеет, сообщу некоторые правила работы с программированными текстами. Чтобы запросить очередной кадр такого текста, нужно нажать клавишу ВВОД. Начинаются все тексты с адреса 1000 и кончаются адресом 37776. В каждый текст можно ввести новый текст: для этого нажмите «?» и ВВОД, а на вопрос «ИМЯ» наберите имя нового текста, закончив его пробелом (обязательно пробелом!), и включите магнитофон. Копирование текста тоже предусмотрено. Для этого нужно нажать НР и ВВОД одновременно, а затем еще раз ВВОД. В режиме «Редактор системы Рига» нажмите ЛАТ, ЗАГЛ, а затем СУ и F одновременно. На вопрос «ИМЯ ФАЙЛА» включите магнитофон на запись, наберите название текста, затем, завершив его пробелом (обязательно пробелом!), произойдет запись текста.

## Кому нужен БК-0010

Зарубежный опыт показал, что первые партии бытовых ЭВМ были приобретены профессионалами — программистами и инженерами. Для них такого рода ЭВМ — новая форма интеллектуального досуга, каким, например, является для многих спорт, радиолюбительство, автомобиль.

Другой категорией потребителей оказались студенты и школьники. Все они обычно начинают с компьютерных игр. К сожалению, многие так и остаются коллекционерами игр, иногда даже превращаются в «игроманов», т. е. людей, проводящих неоправданно большое время за играми с ЭВМ.

Программировать молодежь начинает по различным причинам. Прежде всего это любопытство. Как устроена эта программа? Смогу ли я сделать такую самостоятельно? Смогу ли я сделать эту игру лучше? Сложно ли создать новую игру? — эти и подобные вопросы становятся первым толчком.




ЭВМ обладает редким свойством подчиняться самым изощренным указаниям человека, записанным на ее языке. И после первой простенькой программы возникает неизбежное желание создать более сложную, более «умную», а потом еще более-более. Так незаметно оказываешься вовлеченным в новый, ни на что не похожий мир программирования, в котором постоянно не хватает всего нескольких часов, чтобы найти последнюю ошибку в своей программе. Научившийся программировать быстро начинает извлекать определенные выгоды. Например, если это студент, то он начинает экономить время на выполнении расчетов по обработке лабораторных работ и предварительной проверке алгоритмов решения задач, заданных для больших ЭВМ, к которым нет свободного доступа.

Для школьника БК-0010 может представлять двоякий интерес. Во-первых, именно эта ЭВМ используется в школах при изучении предмета «Информатика и вычислительная техника». Значит, дома можно поупражняться на своей ЭВМ, чтобы практически освоить этот предмет. Во-вторых, ожидается появление большого количества различных программированных репетиторов, тестов знаний, тренажеров, обучающих задачников по разным школьным предметам. С этого момента станет возможной автоматизация учебного процесса в домашних условиях. Кроме того, возможны пакеты программ по определению способностей к той или иной профессии. И наконец, игра. Речь идет не только об играх для детей, примеры которых мы видели по центральному телевидению, но и об обучающих программах для малышей. Такие программы для БК-0010 уже есть, некото-

рые были сделаны автором этой статьи. Программа «Компьютерный мир» позволяет ребенку освоить понятия композиции рисунка, перспективы и составляющих элементов. Все, что требуется от ребенка, — это выбрать элемент на накладной клавиатуре (домик, дерево, фигурка, птица, гора, дорога, озеро, забор или машина), задать место этого элемента на экране, а остальное сделает ЭВМ. В результате работы с подобной программой ребенок без всякой помощи обучается быстро создавать пейзажи на экране. В компьютерных мирах можно задавать простые законы с целью их изучения детьми. Например, в упомянутой программе невозможно поместить дом на небе или солнце на земле. Очень хорошо использовать в детских обучающих программах принцип «оживления» картинок, например собака бегает при нажатии буквы «с», а заяц прыгает при нажатии «з» — в программированном алфавите. Таким образом, в виде игр можно построить обучение арифметике, музыке, рисованию и другим предметам.

Человека, пользующегося дома печатающей машинкой, например литератора, может заинтересовать режим «электронной печати» на основе БК-0010. Известно, что подготовка текстов — самая популярная в мире задача для персональных ЭВМ. Дело в том, что печатающая машинка, подключенная к ЭВМ, приобретает ряд совершенно уникальных свойств. В такой системе текст попадает не сразу на бумагу, а вначале в память ЭВМ. Причем это происходит бесшумно, легким нажатием клавиш. При этом исключено понятие «помарка», так как при любом исправлении буквы она просто замещает старую. Также просто решается вопрос вставки и убирания слов, для этого строку не надо перепечатывать, а достаточно нажатием специальной клавиши подвинуть строку в нужную сторону и вставить слова. Упрощается процедура перемещения абзаца. Для этого вместо перепечатки нужно указать начало и конец абзаца, а затем новое место. Но и это еще не все. Есть программы, автоматически проверяющие правописание слов и даже пунктуацию. Такие программы содержат словари в десятки тысяч слов и сотни правил грамматики. Скорость их работы доходит до тысяч слов в минуту. Наконец, программы, которые называются форматтерами, распечатывают текст с нумерацией страниц, выравниванием левого края текста, центровкой заголовков и другими полезными процедурами, создающими типографское качество печати.

Теперь говорят о возникновении нового направления — домашних информационных центров, которые, будучи подключены через телефон к редакциям газет, журналов и информационным службам, способны формировать интересующие человека материалы прямо на дому. Все это может показаться далеким будущим. Однако это не так. Уже сегодня БК-0010 имеет несколько программных редакторов текста. В частности, есть редактор, позволяющий готовить тексты с произвольными алфавитами, блок-схемами и простыми рисунками. Кстати, эта статья подготовлена таким редактором.

## Программное обеспечение БК-0010

К моменту начала выпуска БК-0010 у нас в стране не было прецедента создания программ для бытовых ЭВМ, поэтому программное обеспечение БК-0010 состояло из транслятора ФОКАЛ и нескольких игровых программ. Однако уже в 1986 г. у некоторых коллекционеров накопилось более 300 различных программ. Что же это за программы? Отметим сразу, что все они — результаты первых попыток разработки любительских программ, поэтому рано еще подходить к ним с общими мерками, тем более что в ряде случаев это были первые программы начинающих авторов.

Все программы можно разбить на четыре группы:

(1) системные, (2) игровые, (3) учебные и (4) расчетные.

Среди системных программ для БК-0010 есть трансляторы языка БЕЙСИК: адаптированный с версии для микро-ЭВМ «ЭЛЕКТРОНИКА НЦ-80» и быстрый транслятор в оригинальной разра-

ботке Вильнюсского университета. Оба транслятора вводятся с магнитофона и могут работать лишь с числовыми данными. В 1986 г. появился вариант транслятора на трех микросхемах (г. Вильнюс). Этот вариант выполнен близко к международному стандарту MSX, что означает возможность использования многих зарубежных программ. Есть попытки разработки и других трансляторов для БК-0010.

К системным программам относится и отладчик программ (г. Рига) в кодах команд БК-0010. Он очень удобен для разработки быстрых и компактных программ на машинном языке, а также как обратный ассемблер программ.

Системной можно считать и разработку ФОКОД (г. Москва), которая позволяет объединять программы на языке ФОКАЛ с программами в машинных кодах.

Известно несколько вариантов редактора текстов, различающихся назначением.

Группа игровых программ, пожалуй, самая многочисленная. Большинство игр написано на языке ФОКАЛ. Одними из первых, получивших распространение, были ХАММУРАПИ, ПОЛЯНКА, МИР (г. Рига). Чаще всего такие программы — результат адаптации программ на языке БЕЙСИК.

Группу учебных программ для БК-0010 правильнее назвать учебно-демонстрационной, так как все программы не отражают систематического подхода к учебному процессу. Это не связанные между собой программы, которые всего лишь доказывают возможность решения отдельных учебных задач. Среди них имеются тесты знаний по разным языкам (русский, английский, немецкий), а также по физике, математике, химии, биологии, истории, географии и другим предметам. Использована и форма тренажера (иностранные слова, физические константы, понятия программирования и т. п.), позволяющая автоматизировать про-

## Стиль программирования

Для каждой группы программ БК-0010 характерны свои особенности, знание которых вырабатывается на практике. Вот некоторые из них, присущие обучающим программам:

— программированный материал в отличие от книги предъявляется кадрами, а не страницами;

— каждый кадр должен быть полным относительно содержащейся в нем информации и не требовать обращения к посторонним источникам или кадрам для его понимания;

— текст кадра должен быть ясным и не перегруженным информацией. Нужно помнить, что средний человек не способен оперировать сразу более чем с 5÷7 новыми понятиями;

— кадр нужно посвящать одной главной для него идее, чтобы упростить последующую задачу контроля понимания кадра;

— все пояснения к кадру должны быть в нем же;

— последовательность кадров должна соответствовать переходу от простого к сложному, чтобы исключить по возможности процедуру возврата, требующую дополнительного программирования;

— норма изложения материала в кадре должна быть диалоговой с обращением внимания на главные детали;

— после каждого информационного кадра должен следовать вопрос на понимание, иначе лучше изложить все в виде книги;

— вопрос должен быть понятным и не требовать дополнительных пояснений;

— вопрос должен выяснять понимание основной идеи кадра. Если одним вопросом задача не решается, то нужно задать последовательно несколько уточняющих вопросов;

— в вопросах следует прежде всего уяснить новые понятия и их связи со старыми и между собой;

— вопрос должен соответствовать кадру и быть полным, недопустимы вопросы типа: «Сколько?», «Как?»;

— анализ ответа должен строиться на поиске важнейших терминов, их число должно быть небольшим (1÷3);

— ответ не должен строиться на долгих операциях (расчетах, поиске литературы и т. п.). Такой контроль проводите устно. Весь анализ должен быть выполним «в уме»;

— на неверные ответы должны быть предусмотрены ясные и полные объяснения, исключающие многократный повтор вопроса;

— вопросы должны требовать краткого и точного ответа. Недопустим вопрос типа: «Объясните...» Такой диалог нужно проводить устно.

Соблюдение указанных рекомендаций позволит вам создавать качественные обучающие программы.

цесс запоминания различных фактов.

Наиболее полной системой является пакет обучающих программ по языку БЕЙСИК (г. Рига). Его цель — помочь тем людям, которые решили заняться самостоятельным изучением этого языка. Пакет содержит три программированных урока по БЕЙСИК, тренажеры по новым понятиям, тесты знаний, задачники и ответы к задачам с рекомендациями по их решению.

Кроме обучения языку БЕЙСИК, подготовлена серия программ по курсу информатики, начиная со знакомства с клавиатурой и возможностями ЭВМ: ПЕРЕКРЕСТОК, ПОЖАР, ИЗМЕРЕНИЯ, ПЕЙЗАЖ, ГОРОДА СССР, КЛАВИАТУРА (г. Рига). Эти программы позволяют не только последовательно освоить клавиатуру БК-0010, начиная с клавиши ВВОД, но и показать возможности использования БК-0010 в самых разнообразных ситуациях. Кроме программ первого контакта с ЭВМ, имеются описания и задачники к учебному редактору текстов и базе данных. В качестве примера подготовлены две базы данных — ЕВРОПА и ЗНАКОМЫЕ.

Основные понятия программирования: «команда», «серия команд», «повтор», «цикл», «метка», «переход по метке» — поможет освоить программа — исполнитель РОБОТ, в которой можно управлять движением человечка-робота по комнатам.

Очень интересна немногочисленная группа учебных моделей. Некоторые из них перекочевали в БК-0010 из описаний зарубежных программ. Такова, например, удачная программа МЕНЕДЖЕР (г. Москва), в которой обыгрывается простая экономическая модель управления фирмами. К этой же серии относится упомянутая выше ХАММУРАПИ, в которой предлагается управлять хозяйством древнего государства. Остается надеяться на появление таких же увлекательных программ-моделей колхоза, завода, кооператива и т. п., чтобы можно было, пусть в игро-

## Задачи для БК-0010

Количество задач, которые применимы для БК-0010, неограниченно. Однако для новичка их поиск может стать проблемой. Ниже перечислены некоторые из задач, актуальные в первое время общения с ЭВМ.

1. Рисование по экрану с сохранением результата на магнитной ленте. Желательно иметь несколько «кисточек» разного цвета и разной толщины управляемых как курсор. Нужен редактор рисунков, позволяющий «вырезать участки, смещать их и налагать друг на друга.

2. Исполнение мелодий. Требуется редактор с прямым вводом нотных обозначений на нотный стан и немедленным исполнением как по отдельным звукам, так и целиком. Исполнитель должен иметь несколько тембров и октав и набор специальных звуковых эффектов.

3. Электронная таблица. С ее помощью нужно задавать структуры и сами операции в каждой позиции, а также возможность связи позиций между собой. Должно быть также предусмотрено обращение к элементам путем установа в них курсора.

4. Анализаторы текстов. Годны программы различных применений. Например, поиск грамматических ошибок, упорядочение слов вводимого текста по разным признакам (алфавитным, буквосочетаниям, слоговым и т. п.), подсчет слов, буквосочетаний, букв в тексте по заданным признакам и т. п.

5. Генераторы игр. Такие программы способны формировать декорации игр, движущиеся объекты, правила движения и логику действия объектов и стратегию выигрыша.

6. Генераторы программированных фильмов. Все аналогично генераторам игр, но добавляется сюжет и его исполнитель во времени. Оформление должно быть звуковым и с субтитрами. Особенно интересна возможность диалога во время фильма.

7. Программированный туризм. Аналогично 6, но в качестве материала берутся города, дороги между ними или фантастические пейзажи. «Путешественник» может перемещаться в запрограммированном пространстве.

8. Программированный атлас. Речь идет о картах, введенных в ЭВМ, которые можно запрашивать в разных масштабах.

9. Преобразователи текстов, вводимых с пульта в речь ЭВМ.

10. Проблемно-ориентированные трансляторы для учебных предметов. Имеется в виду создание языков и программных средств к ним аналогично тому, как это было сделано для специалистов-физиков (ФОРТРАН), экономистов (КОБОЛ) и др., но в данном случае — для учителей химии, истории, литературы и т. п. Примером такого языка может быть Т-язык.

11. Пакет обучающих программ. Здесь возможности неограниченны. Актуально все, что представляет интерес для самообучения: языки (иностранные и искусственные), машинопись, быстрое чтение, школьные предметы и вузовские курсы, устройство ЭВМ, правила дорожного движения и т. п.

12. Домоводство. Это может быть расчет семейного бюджета, рецептов блюд, количества удобрений, вносимых в огороде, и управление электронной сигнализацией в доме, а также бытовыми приборами.

вой обстановке, ощутить специфику принятия решения по экономическим и организационным вопросам этих предприятий.

К учебно-демонстрационным программам примыкают и просто демонстрационные. Например, программы ДОМИК и СЧЕТ (г. Рига) показывают возможность использования программированной речи при обучении детей рисованию и счету. Программы ПЕСНИ, ВИКТОРИНА (г. Рига) и МЕЛОМАН (г. Москва), возможно, вызовут интерес у учителя музыки и у любителей компьютерной музыки.

Группа расчетных программ БК-0010 особенно разнообразна. Здесь есть и программы выполнения узкого анализа, но секрет назначения знают лишь авторы. Большая группа программ посвящена решению уравнений, интегрированию и статистике. Есть и медицинские программы.

## Надежды и возможности

Излишне оптимистическая (а иногда пессимистическая) информация по отношению к персональным ЭВМ вообще и к БК-0010 в частности — источник многих заблуждений и разочарований новичков. Нередко в прессе можно прочесть, что БК-0010 может заменить телефонную книгу, готовить пищу, обучать любым предметам и т. п. Автору приходилось встречаться с людьми, которые приобрели БК-0010, чтобы, например, автоматизировать систему выплаты зарплаты или сделать информационно-поисковую систему по научным публикациям, или создать банк данных по уголовному кодексу и т. д. Приходилось объяснять, что вся память БК-0010 — это всего лишь несколько листов печатного текста. А при таких объемах лучше информацию просто записать и просматривать вручную. Иное дело — расчетные или динамические программы, которые порождают информацию на сотни и тысячи страниц.

С другой стороны, есть люди, которые считают, что БК-0010 нужно немедленно снять с производства, так как она совершенно ненадежная, без программ, с клавиатурой, на которой невозможно работать, и с телевизором, на который вредно смотреть. Такую точку зрения обычно имеют люди, либо не разобравшиеся в чем дело, либо просто недоброжелатели.

Конечно, вышеперечисленное не снимает с повестки дня вопрос о сети сервисных служб по бытовым ЭВМ, где могут не только быстро и качественно устранить неполадки в БК-0010, но и оказать необходимую консультацию по обслуживанию и программированию ЭВМ, подключению к телевизору и магнитофону.

Приход ЭВМ в семью — несомненно, новое и сложное социальное явление, поскольку оно затрагивает информационную структуру людей. Сейчас трудно сказать, к каким последствиям это приведет даже в обозримом будущем, но несомненно, что эти последствия могут быть как положительными, так и негативными, если все пустить на самотек. Здесь можно провести параллель с возникновением и распространением кассетных магнитофонов. Магнитофонные записи сделали возможным приобщение самых широких масс к современной музыке по выбору. Собираются дома гости. Какую желаете песню? И пожалуйста, слушайте! Но, увы, как редко теперь слышатся песни самих гостей и как часто звучат записи сомнительного качества и содержания. Программы бытового компьютера имеют много общего с музыкальными записями, они также распространяются на кассетах, их можно копировать в домашних условиях и делать тематические подборки. Содержанием программ являются обычно тексты и картинки, выдаваемые ЭВМ на телевизор, а способ воздействия — диалоговый. При прослушивании музыки нужно лишь вставить кассету и включить кнопку воспроизведения. При работе с ЭВМ нужно постоянно отвечать на ее вопросы, ибо здесь режим работы активный.

Итак, процесс всеобщей компьютеризации начался, приглашайте ЭВМ в семью и не бойтесь ее. Она способна стать вашим другом и обогатить вас новыми знаниями.



**Предлагаем в виде самосовершенствования заняться на компьютере делом и занимательным, и вполне серьезным — моделированием некоторых простейших физических (да и не только физических) явлений. Ведь эта сфера и сейчас остается, пожалуй, самой важной и емкой областью использования компьютеров. Заведя себе ПЭВМ, было бы просто нелепо не попытаться составить представление о столь важном направлении. Глядишь, удастся оживить и заложенные некогда в глубины мозга сведения из школьной и институтской физики и математики...**

Н. ГЕРМАН

# ВСЕ — ЗА КОМПЬЮТЕРЫ!

С момента появления в широкой продаже первого отечественного персонального компьютера БК-0010 прошло не так много времени, но уже можно заметить, что спрос на это чудо техники после периода начального ажиотажа заметно упал.

Более всего здесь настораживает активность рекламы, которая у нас выполняет всегда довольно своеобразные функции. Может внушить беспокойство и пустынный зал магазина «Электроника», где два скучающих продавца вечно режутся в игры на БК... А вот оживление у прилавков магазина «Вычислительная техника», где торгуют калькуляторами, никогда не стихает. За рубежом, однако, уже успело прокатиться несколько волн спроса на ПЭВМ.

Уместно поэтому задаться вопросом: для чего вообще необходимы персональные компьютеры (ПЭВМ)? Конечно, если по роду своей профессиональной деятельности вам приходится обращаться к расчетам на ЭВМ, ответ может быть достаточно ясным: теперь, как может показаться, все подобные расчеты можно делать дома.

Ну а если вы не профессиональный математик, физик, программист и т. п.? Попробуйте сообщить своим знакомым, что купили или еще только намереваетесь купить себе персональный компьютер, и, скорее всего, услышите в ответ: «А какие в нем есть игры?» Итак, ПЭВМ — просто очень сложная игрушка? Что и говорить, компьютеры сумели изрядно преуспеть и в игровом деле. Известно, правда, что в шахматы хороший мастер пересилит самую сложную ЭВМ, но вот, например, в нарды машины уже более десятка лет обыгрывают даже чемпионов мира (существуют и такие). Одним словом, из нашего БК вполне может получиться достойный партнер, крупье, а возможно, и... шулер. Итак, компьютеризируем домино, очко, лото, преферанс?

Символы карточных мастей на клавиатуре БК говорят о том, что конструкторы не обошли вниманием это важное направление развития человеческого духа. Отметим, что игра, записанная на штатной кассете БК под девизом KOSTI, представляет собой точное воспроизведение известной и до компьютеров игры, где партнеры добывали себе очки именно бросанием костей, что на БК «моделируется» производством случайных (а фактически, как мы скоро убедимся, псевдослучайных) чисел. В качестве курьеза укажем, что в журнале «Радио» (1985, № 6) предлагалось использовать микрокалькулятор для «угадывания» результатов «Спортлото» (также путем генерации случайных чисел). Программа бросания костей на N граней для калькулятора приведена в апрельском выпуске нашей серии (см. Отечественные калькуляторы).

Впрочем, оставим эту тему для отдельного обсуждения.

Так чем же посерьезнее можно заняться на БК? Пробежимся еще раз по руководству пользователя и просмотрим рекламный ролик, записанный на штатной кассете БК. Ведение семейного бюджета, регулирование микроклимата и освещенности в квартире, создание собственного банка данных — все это звучит заманчиво. Итак, заводим этот самый банк. Но вот что в нем хранить? Каталоги коллекций марок? Варианты обмена квартиры? Телефоны знакомых? Досье разве на кого завести? М-да, кое-какие возможности, конечно, вырисовываются. Но нужно прежде всего выяснить, какие преимущества мог бы иметь компьютерный банк, основанный на БК, по сравнению с обычным блокнотом, скоросшивателем или амбарной книгой. Хранимые данные необходимо прежде всего «напечатать», т. е. набрать на клавиатуре

БК, затем переписать их на магнитную ленту, а при необходимости прочтения или переработки — перевести обратно с ленты в машину и т. д. Достоинство электронных банков — в возможности быстрого автоматизированного поиска, обработки, перетасовки информации. Если же при этом не организовать автоматизацию всех процессов записи, перемоток и поиска нужных участков ленты на магнитофоне, то практически все мыслимые преимущества нашего банка улетучатся.

Последние модели магнитофонов в принципе позволяют (при доработке) автоматизировать режим перемотки, а некоторые — и ускоренный поиск нужного участка. Дело это, однако, по плечу лишь опытным радиолюбителям. Специализированных магнитофонов, а тем паче дисководов, наиболее приспособленных для хранения данных, в продаже пока нет.

С ними испаряются и надежды на поправку семейного бюджета за счет БК, поскольку его роль в этом деле реально может свестись разве что к аккуратной записи расходов. Регулировать освещенность и квартирный микроклимат, разумеется, принципиально возможно, но цель здесь вряд ли оправдает затраты интеллектуальных и материальных ресурсов (ведь понадобятся какие-то ЦАП и АЦП и т. п.) да и старые средства — шторы, выключатели и форточка — не кажутся пока безнадежно устаревшими...

Микросхемы ЦАП и АЦП теперь свободно можно купить в той же «Электронике». Но в целом предлагаемое применение БК кажется малопрактичным и вообще надуманным.

В серии статей, начатой в журнале «Энергия» (1986, № 8), мы предложили новым рекрутам кибернетизации для освоения возможностей БК и в виде самосовершенствования заняться на компьютере делом и занимательным, и вполне серьезным — моделированием некоторых простейших физических (да и не только физических) явлений.

Дети вот поглядывают на компьютер: игры-играми, но их тоже пора знакомить с существом дела... Словом, вперед, за моделирование!

С чего же начать в этом направлении? Любая задача, связанная с моделированием какого-либо процесса, явления, всегда может быть разложена на ряд элементарных этапов — от представления нужной информации в привычном виде, нахождения точек пересечения каких-либо зависимостей, т. е. решения линейных или даже трансцендентных уравнений (систем уравнений), до построения решений дифференциальных уравнений в частных производных (это и самая общая форма выражения наиболее фундаментальных законов природы, и самая, пожалуй, трудная в практическом отношении проблема). Обычный путь постижения любой науки — движение от простого к сложному. Этот путь не короток, но нам, конечно же, придется так или иначе следовать подобному восхождению к сложности. Но начать мы предлагаем ни со сложного, ни с простого, а со случайного.

В кибернетике известен такой тезис: всякий выбор, отличный от случайного, порочен. Автор, выбирая «случайное» начало, руководствовался не этим экстремистским лозунгом, но более рационалистическими соображениями...

Полагаем, что таким путем нам удастся гораздо быстрее окунуться в гущу премудростей моделирования и ощутить кое-какие прелести и ограничения (это не менее важно) компьютера. Вообще это, пожалуй, лучше всего соответствует духу названия всей нашей серии. За рога — так за рога!

Автор имеет в виду рубрику

**БК·ЗА·РОГА**

в которой будет рассказано о работе с компьютерами БК-0010 и БК-0011. Статьи этой рубрики см. в последующих выпусках.

Набираем программу:

```
1.1 S  X=128; S  Y=128
1.2 X  FT(1,X,Y)
1.3 S  X=X — FRAN (   );
    S  Y=Y — FRAN (   )
1.4 G 1,2
```

Большая часть распроданных БК понимает ФОКАЛ, и поэтому мы будем печатать программы в этом варианте. Для пользователей БК с БЕЙСИКом мы будем указывать смысл операторов, их аналоги и т. п. Перевод наших достаточно простых программ — дело несложное, тем более что оба языка восходят к общему источнику — ФОРТРАНу. Как мы увидим, ФОКАЛ, представляющий собой как бы предельно

упрощенный диалект (своего рода пиджин-инглиш), создает своему пользователю определенные удобства: операторы обозначаются всего одной буквой, строки программы можно легко объединять в страницы и т. д. Но разумеется, у более развитого БЕЙСИКа свои серьезные преимущества.

После ее запуска в центре экрана возникает точка, которая хаотически мечется по экрану, зачерчивая за собой причудливую область, заполняющую постепенно все большую площадь. Какое же физическое явление мы здесь смоделировали? Оказывается, это неплохая модель броуновского движения — хаотического перемещения мелких частиц под действием теплового движения молекул. Явление было случайно подмечено английским ботаником Р. Броуном (теперь бы он был, скорее, Брауном) в 1827 г. при наблюдении пыльцы растений в микроскоп. Это явление изучалось в течение не менее столетия как самим Броуном, так и другими именитыми учеными и сыграло важную роль в окончательном становлении атомистических представлений.

В нашей модели хаотические перемещения частицы задаются особой случайной функцией FRAN (), которая каким-то образом заложена (как говорят еще, «защита») в памяти (ПЗУ) БК. Со свойствами нашей функции FRAN и других случайных функций нам еще предстоит основательно освоиться. Пока же обратим внимание на то, что наблюдателю след частицы, разумеется, не виден. Как его убрать с экрана? Добавим строчку:

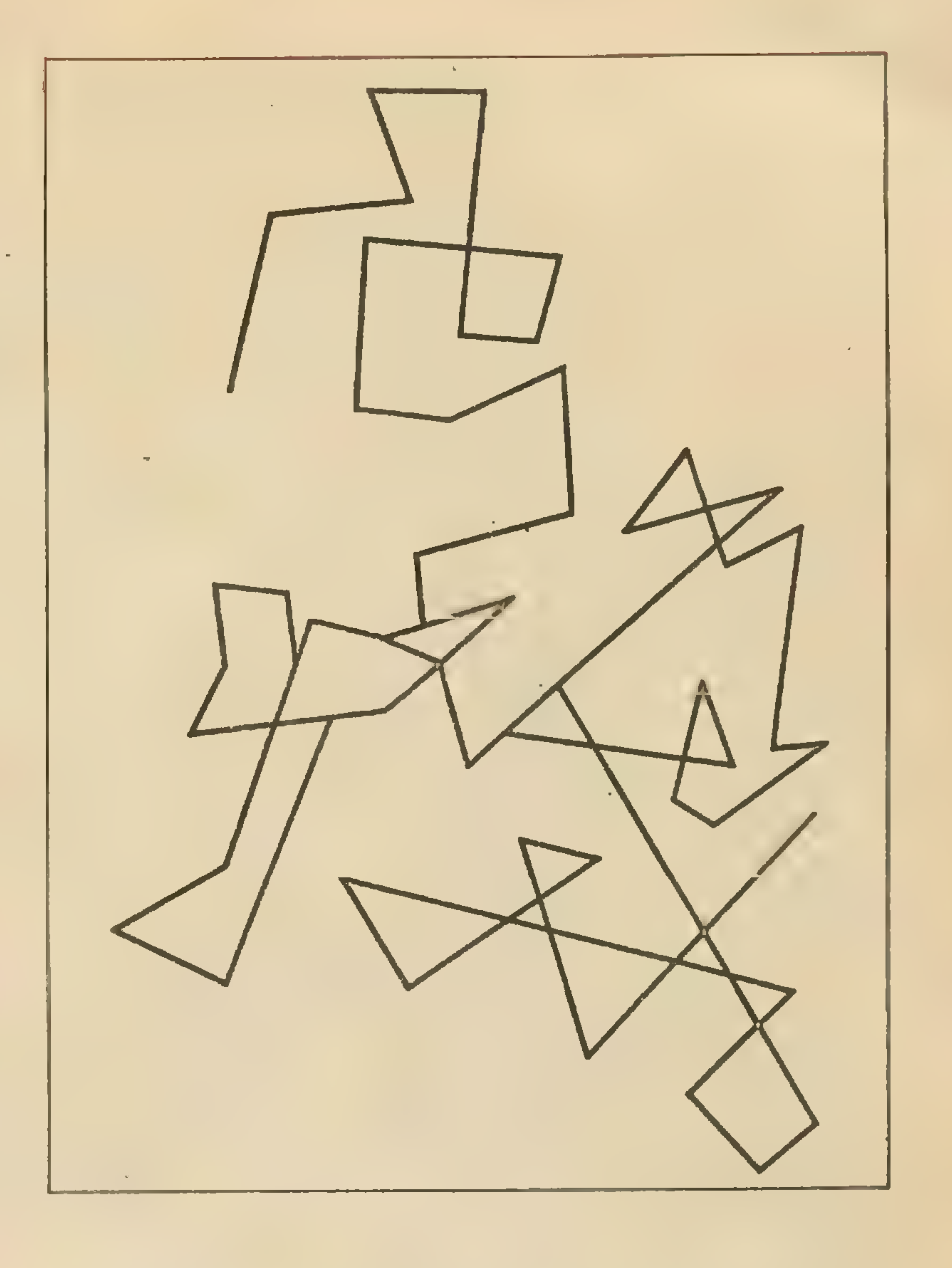
```
1.25 X  FT (Ø, X, Y)
```

След пропадает, но наша точка теперь принимается мигать, напоминая не броуновскую частицу, а, скорее, порхающего мотылька. Можно попытаться слегка приглушить это мигание:

```
1.25 X  FT (Ø, A, B)
1.27 S  A=X:S  B=Y
```

Идею можно обобщить, что приведет к ползанию червяка или амебы длиной *K*:

```
1.05    A „K“, K
1.10    S  X(K+1)=128;   S  Y(K+1)=128
1.15    F  N=1, K;  SX(N)=X(N+1):S  Y(N)=4(N+1)
1.20    S  X(K+)=X(K+1)+FRAN(   ); S  Y(K+1)=Y(K+1)+FRAN(   )
1.30    X  FT(1, X(K), Y(K)); X  FT(0, X(1), Y(1)):G 1.15
```



Оператор F — то же, что FOR БЕЙСИКа, его первый операнд — начальное, последний — конечное значение параметра цикла, средний (если он указан) — шаг цикла. Если его нет, шаг равен единице. Оператору A(ASK) в БЕЙСИКе соответствует INPUT.

Чтобы оценить заслуги классиков науки, попробуем, как это производилось при экспериментальном изучении явления, зарисовать след частицы (он именуется броуновской траекторией), например, наложив на экран кальку. Возможности документальной записи информации с экрана БК и по сию пору остались столь же ограниченными, как и в самом начале его производства. О графопостроителе не приходится и мечтать. Многие любители сопровождают сообщения в печати (см. «Науку и жизнь») о собственных изысканиях фотографиями с экрана, что говорит о готовности предпринять ради любимого дела самые героические усилия. Тем временем по слухам кто-то где-то изобрел процесс мгновенной съемки изображения с экрана на некую пленку, накладываемую непосредственно на экран. Не станем унывать, глядишь, в резуль-

тате графопостроитель станет вообще ненужным.

Положение точки при этом следует фиксировать через равные интервалы времени. Чтобы не отвлекаться, устроим себе «часы» прямо в углу экрана:

1,35 X FK(0,0); T %5, T; S T=T+1

Такой счетчик показывает не мировое время, а лишь число «шагов» БК, но пока нам допустимо считать, что он «шагает» равномерно, что, конечно, не совсем точно.

Оператор X FK (X, У) устанавливает начальное положение курсора, откуда и начинается печатание. Число, стоящее за знаком %, определяет полное количество знаков печатаемой величины (целая часть) и количество знаков после запятой (дробная часть).

Чтобы нащупать наиболее благоприятные условия наблюдения броуновских траекторий, можно менять величину элементарного шага частицы.

Предложенное положение центра экрана соответствует укрупненному масштабу по оси X, более удобному для пользователей с недостаточно острым зрением (увы, и автору). Переключение масштабов производится клавишей 32/64, нажимаемой вместе НР. Код этой команды — 155. Действие клавиш БК понимает даже во время выполнения какой-либо программы. Те же особенности имеет и команда «курсор» (код — 154), включающая и выключающая надоевший мельканием курсор.

Увеличим его, например, в 5 раз (тем самым мы смоделируем переход к большему увеличению в нашем «микроскопе»). Последовательные положения частицы при зарисовках принято соединять отрезками прямых линий (это не означает, конечно, что частица действительно двигалась по прямым). Восстановим след:

1.30 S X=X+5*FRAN(); S Y=Y+5*FRAN()
1.12 X FT(0, X, Y); X FV(1, X, Y)

В таком виде след частицы на экране напоминает результаты непосредственных зарисовок исследователей, которые можно найти хотя бы в БСЭ. Но наш БК позволяет заметно облегчить процесс зарисовки траектории, поскольку вся она сохраняется на экране и ее можно срисовать в конце процесса целиком, не гоняясь за самой частицей.

Какие же цели преследовались при подобных зарисовках, на которые пришлось потратить немало терпения и изобретательности? Оказывается, при всей явной запутанности траекторий в них может быть выявлен целый ряд своеобразных статистических закономерностей. С некоторыми из них нам еще предстоит ознакомиться. Пока же мы предложим нашим читателям несколько упражнений для более основательного знакомства как с самими случайными блужданиями (таково математическое наименование подобных процессов), так и с возможностями нашего премудрого БК. Полагаем, что и сами читатели смогут разработать какие-то свои варианты наблюдения случайных процессов.

1. Траектория приближается к краю экрана и, того гляди, вообще уйдет за него. Заставьте ее «отразиться» от этого (вообще любого) края, а затем и от любой заданной границы, причем не обязательно прямолинейной. Не сможете ли вы заставить частицу блуждать внутри какой-либо окружности?

Пока нас мало заботит качество отражающей границы. Ведь частица всегда оказывается от нее на случайном расстоянии. Достаточно, например, такого простого устройства:
1.4 I (Y) 1.5, 1.5, 1.2
1.5 S Y=Y+1; G 1.2

Оператор I это IF и в БЕЙСИКе, но адресация у него иная: первым указывается адрес, куда следует обратиться, если выражение, следующее за I, отрицательно, вторым, если оно равно нулю, затем, если оно положительно. Если адресов не хватает, будет выполняться следующая строка.

2. Пусть частица, исчезнув с одного края экрана, «возникнет» с противоположного края (сверху — снизу).

3. Выпустите из центра одновременно несколько (две, пять, сто) частиц. Выпустите несколько частиц из разных точек экрана.

4. Заставьте ваш БК погонять частицу в течение некоторого достаточно большого времени (более 40 тыс. шагов). При этом придется предусмотреть какие-то меры, чтобы частица действительно не уползла за край экрана, — подобрать масштаб, может быть, устроить отражающие границы. Поскольку блуждающая точка часто теряется в уже зачерченных ею областях, можно научить ее подмигивать один или несколько раз на каждом шагу или после определенного числа шагов.

При выполнении любых упражнений, требующих длительного наблюдения, телевизор (после того как вы убедились, что программа начала функционировать нормально) можно использовать по его прямому назначению (во избежание косых взглядов домашних на всю эту кибернетику). Вырабатываемое БК изображение будет при этом храниться в его памяти (ОЗУ).

Подобные длительные «зарисовки» полезно проделать и с несколькими частицами, выпущенными из разных точек экрана (постаравшись, чтобы траектории не перепутывались слишком сильно).

При достаточно большом числе шагов можно будет обнаружить, что в изображаемых фигурах начинает проступать определенная симметрия. Для того чтобы выявить ее более наглядно, повернем наши фигуры на 45°. Для этого достаточно откладывать на координатных осях не сами переменные X и У, а значения X — У, и X + У (при этом изменится, правда, и масштаб, но при желании и это можно подправить, введя дополнительный масштабный множитель $1/\sqrt{2} \simeq 0.7$). Предупредим сразу возможное недоумение наших исследователей: реальная броуновская траектория, разумеется, не может приобрести никакой симметрии, это было бы слишком случайно! Наш результат — первое свидетельство неких особенностей процесса генерации «случайных» чисел внутри БК.

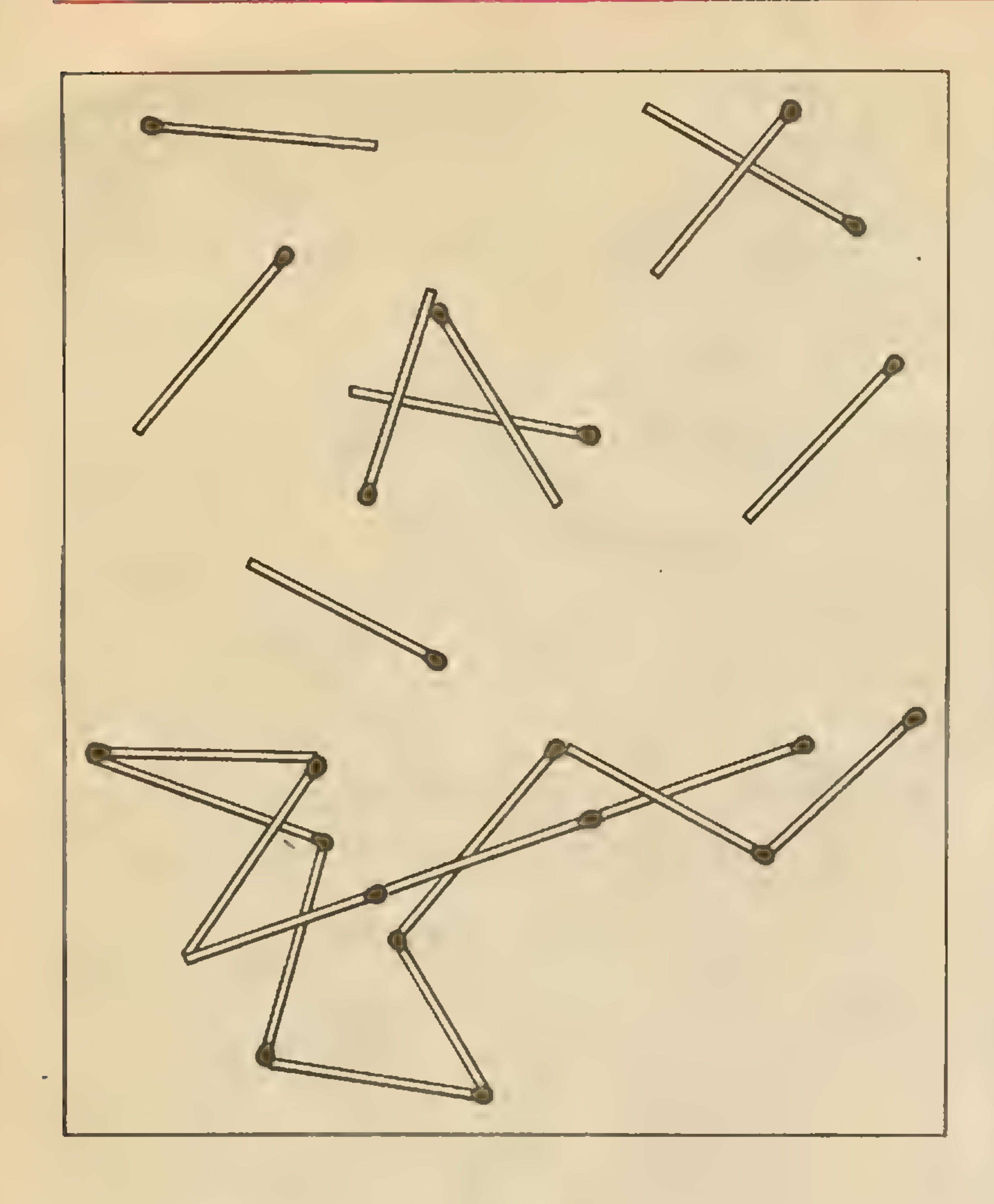
5. Все предыдущие упражнения можно видоизменить, заставив частицу совершать шаги постоянной (например, единичной) длины вдоль каждой координатной оси. Полезно отметить, как трансформируется вид фигуры, получаемой при большом числе шагов, в этом варианте. Не сохраняется ли все же определенное сходство?

Еще одно полезное усложнение — организовать перемещение шагами постоянной длины, но ориентированными случайным образом. Наконец, почему бы не заставить нашу частицу перемещаться не по плоскому экрану (т. е. всего в двух измерениях), но подобно тому, как блуждает реальная броуновская частица, — в трехмерном пространстве? Ну а в четырех-, пятимерном? Конечно же, в конце концов результат должен быть поневоле представлен в виде проекции все на тот же плоский экран, но не возникнут ли при этом какие-то отличия?

Пока компьютер предается заданным вычислениям, пользователи получают блестящую возможность поразмышлять на сопутствующие темы (одна из возможных тем о пользе компьютеров, роботов и прочих автоматов как средств, высвобождающих человеку время для размышлений). Подкинем им некоторые сведения из теории случайных блужданий для постепенного осмысления.

Основной закон, который выверялся и при экспериментальном изучении броуновского движения, заключается в том, что частица в среднем удаляется от любой начальной точки таким образом, что средний (т. е. усредненный по множеству независимых испытаний) квадрат удаления частицы от этой точки растет пропорционально времени. Для случая же, когда, как и на экране БК, блуждания осуществляются конечными дискретными шагами, средний квадрат удаления $\overline{\Delta r_N^2}$ оказывается просто равным $N\overline{\Delta r_1^2}$, где $N$ — число шагов, а $\overline{\Delta r_1^2}$ — средний квадрат длины одного шага. Приведем пример, который может пояснить существо этого важного закона.

Бросим на пол по возможности наиболее беспорядочным образом $N$ спичек, а затем сложим все спички по правилам сложения векторов. Какую длину будет иметь результат такого сложения? Может показаться, что он должен равняться нулю, поскольку все направления на плоскости равноправны и «результат» просто некуда направить. Нетрудно представить, однако, что, для того чтобы обратить сумму всех наших векторов в нуль, придется как-то весьма специальным образом разложить их на плоскости. Поэтому какая-то случайная «некомпенсация» всех направлений всегда образуется, причем в среднем результат (сумма длин спичек как единичных векторов) будет равен $\sqrt{N}$. Сами же средние смещения (т. е. вектор результирующего смеще-



ния снова усредненный по многочисленным испытаниям) должны равняться нулю именно в силу изотропии, т. е. равноправия всех направлений на плоскости.

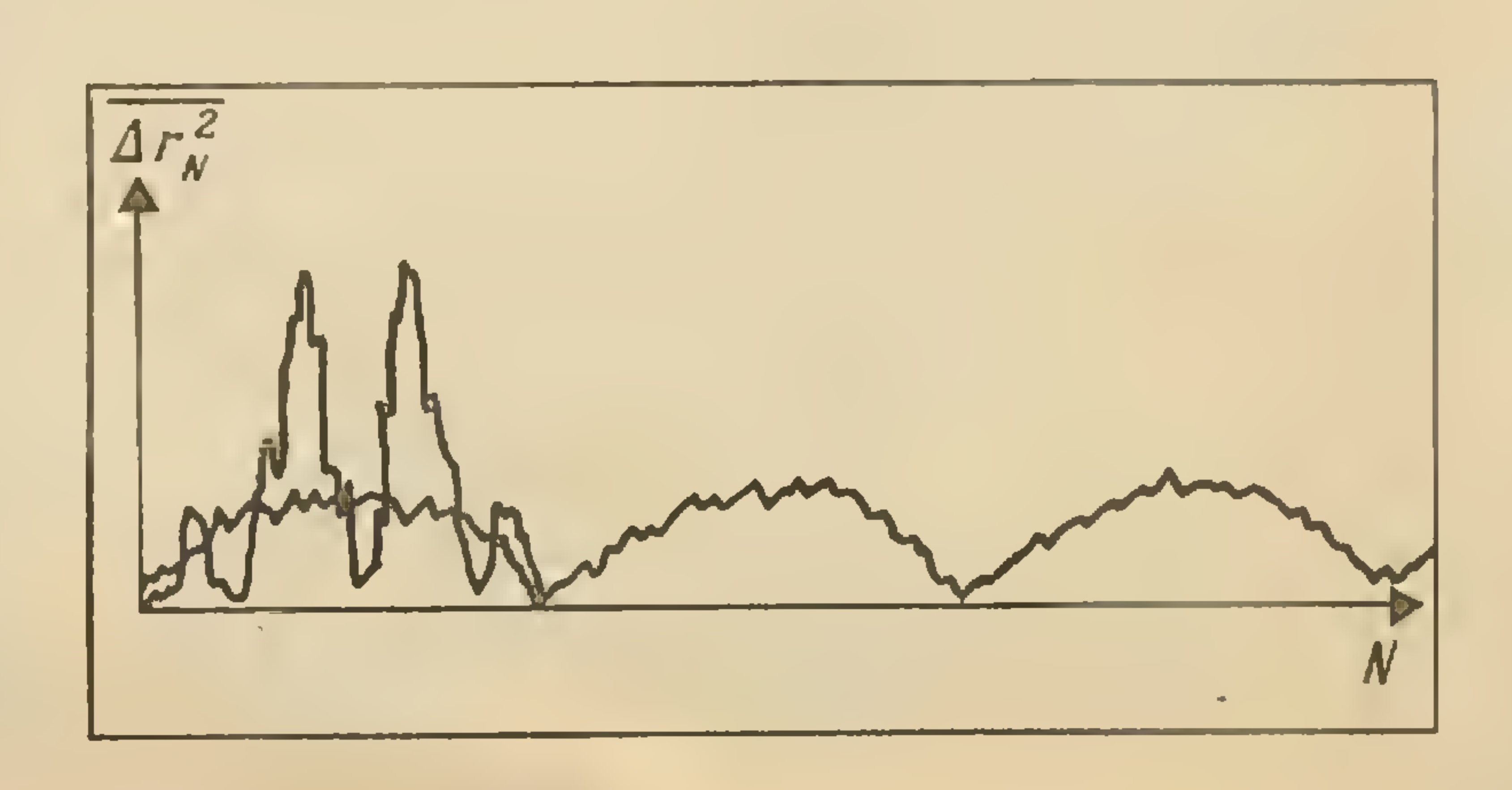
«Спичечной» интерпретацией удобно воспользоваться, чтобы самостоятельно доказать основной закон, хотя бы методом математической индукции. Полезно попытаться вычислить и средний квадрат длины одного шага.

Как организовать проверку этого основного закона для моделируемого нами движения? Самое простое — печатать вслед за показаниями «часов», т. е. числом шагов и квадрат величины $\Delta r_N^2$. Это стоит проделать хотя бы для того, чтобы убедиться, что вот так просто, на глазок, никакой вообще связи $\Delta r_N^2$ и N заметить невозможно. Дело здесь в том, что любой подобный статистический закон, утверждающий что-либо о связи средних величин, еще не вполне полон, пока он не сопровождается указаниями о допустимых отклонениях от этих средних, или, выражаясь научным языком, о дисперсии (разбросе) результатов. Так вот оказывается, что для случайных блужданий по плоскости дисперсия этого самого квадрата удаления непривычно велика (она просто равна самому среднему значению), что порой ускользает даже от внимания специалистов. Дабы в этом убедиться, советуем сопроводить блуждания нашей частицы графиком зависимости $\Delta r_N^2$ от $T$, для чего придется подобрать наиболее удобный масштаб, в котором следует откладывать на осях $X$ и $У$ наши переменные. Обычно для выбора масштаба любой покуда еще неизвестной зависимости уважающий себя экспериментатор справляется у знакомого теоретика: что должно получиться? При этом на осях стараются строить не сами измеряемые переменные, но какие-то их функции с тем, чтобы ожидаемая теоретическая зависимость была прямолинейной.

Этот прием весьма полезен, поскольку разглядеть образующуюся из точек прямую очень просто. Наклонив график и посмотрев вдоль прямой, можно немедленно выявить и характер отклонения результатов от «теории», что может дать повод к выявлению причин расхождений. Наконец, наклон прямой и отрезки, которые она отсекает на осях, также можно немедленно сравнить с предсказаниями. Если же получающаяся кривая — именно кривая, то убедиться, что она «правильная» гораздо сложнее.

Мы с вами, уважаемые пользователи, находимся здесь в весьма выгодной позиции: связь $\Delta r_N^2$ и $T$ известна из теории, и эта связь линейна. Итак, вперед, за построение графиков!

Следует подчеркнуть, что возможность построения различных графиков с помощью БК — необычайно благодарное занятие, и в этом деле пользователь обязан как следует набить себе руку. Правильный подбор масштаба, изображение масштабных меток или сетки, размещение дополнительных надписей, обозначение различных серий экспериментальных точек и т. д. — вот направления неограниченных возможностей совершенствования. На нашем графике можно будет найти место и для изображения самой траектории, но его можно и изъять, если



необходимо добиться максимальной скорости изысканий. Дело в том, что нелишне наш график продолжить и до 100 тыс. шагов. Для этого можно заставить БК поработать всю ночь, ведь ЭВМ не нуждается во сне. Единственно, за чем нужно при этом проследить, это чтобы ни БК, ни его источник питания не были завалены бумагами, а находились бы на открытом месте. Надеемся, что, вскочив утром в радостном волнении истинного исследователя и включив телевизор (именно его, пожалуй, и не следует держать включенным всю ночь), вы обнаружите, что ожидаемая теоретическая прямолинейная зависимость... не выполняется. Что же это такое? Снова шутки дисперсии?!

Дисперсия, конечно, дисперсией, но с ней, оказывается, можно побороться. Наиболее дотошные читатели могут даже по этому поводу вычислить дисперсию квадрата удаления и обнаружить, что... в пространстве размерности $K$ нормированное значение дисперсии равно $\sqrt{2/K}$. Самое интересное, пожалуй, в том, что эта величина не зависит от числа шагов. Поэтому обычные в статистике надежды на то, что чем больше шагов, тем лучше все получается, в данном случае не работают. Увеличивать надо не число шагов, а размерность пространства.

Мы не случайно уже предлагали нашим пользователям поупражняться в блужданиях в многомерных пространствах. Нетрудно сообразить, что каждое последовательное обращение к функции FRAN вполне эквивалентно невидимому шагу в неком невидимом измерении. Просуммировать же квадраты смещений по всем двумстам измерениям для БК — сущий пустяк.

Нетрудно сообразить, что запуск большого числа частиц и суммирование (непосредственное или просто графическое — путем наблюдения за площадью, зачерчиваемой всеми частицами) квадратов их удалений также соответствуют увеличению размерности броуновского пространства. Можно использовать и одну-единственную плоскую траекторию, суммируя квадраты удаления не от одной исходной, а от большого количества последовательных точек. Этими приемами можно основательно сгладить первоначальную довольно корявую кривую.

Зато дисперсия в таком пространстве составит всего 10%, и отклониться теперь от ожидаемой прямой заметным образом наши результаты не должны... Не должны-то не должны, а вдруг? Но оставим на этом месте самых дотошных читателей, если они пожелают еще глубже ознакомиться с особенностями случайных блужданий. Предложим им напоследок такую вот тему для самостоятельного исследования: можно ли на БК продемонстрировать, что основной закон случайных блужданий выполняется с точностью, с которой он был экспериментально проверен для случая броуновского движения, т. е. в несколько процентов? Здесь снова придется поразмышлять над терпением классиков науки.

Именно путем обработки броуновских траекторий было впервые измерено значение постоянной Больцмана $k$, а через него и число Авогадро $N=R/k$ ($R$ — газовая постоянная), т. е. вес атома. Рекордная для своего времени точность определения была достигнута при наблюдении броуновского движения крутильного маятника, фиксировавшегося, правда, киносъемкой.

Нам же настала пора более пристально приглядеться к свойствам нашей случайной функции — у нас уже были основания заподозрить, что с нею что-то не так... Что же именно? Давайте для начала проверим, действительно ли FRAN производит случайные числа, равномерно распределенные в интервале (−1, +1). Для этого предлагаем построить такую гистограмму:

```
1,1 F M=1,50; X FT (1, X(M), Y*M+25); SA=25*(FRAN()+1); SX(A)=X(A)+1
1,2 G 1,1
```

Здесь мы разделили весь рабочий интервал на 50 частей. При попадании очередного случайного числа в какой-либо интервал высота соответствующей клеточки гистограммы (а гистограмма — это «клеточный» график) увеличивается на единицу. Со временем, правда, возникнет опасность, что клеточки выползут за край экрана. Можно поэтому построить зависимость величины реального заполнения каждого интервала от ожидаемого (среднего) значения:

```
1,1 F M=1,50; X FT(1,X(M)*128/(T+1), 25+Y*M) и т. д.
1,2  X  FCHR(12); X FK(0,0); TT; ST=T+1; G 1.1
```

Проследите, как меняется разброс (дисперсия!) от среднего заполнения при увеличении количества чисел. Хорошая случайная последовательность должна обеспечивать не только равномерность заполнения рабочего интервала, но непременно и соответствие дисперсии заполнения своему теоретическому значению. Напомним (а это один из самых важных и наиболее практически популярный результат теории вероятностей), что для малых интервалов при среднем заполнении интервала дисперсия заполнения равна $\sqrt{N}$.

Распределение отклонений от среднего $\overline{N}$ при больших $N$ стремится к известной функции Гаусса:
$P(N)=\exp(-(N-\overline{N})^2/(2\overline{N}^2)$.
Такое распределение называется нормальным, а приведенное утверждение преподносится как выражение закона больших чисел. Если попытаться проверить выполнимость этого закона для FRAN, то может возникнуть проблема правильной нормировки функции Гаусса. Укажем, что интеграл по $N$ от $-\infty$ до $+\infty$ от приведенного выражения равен $\sqrt{2\pi}\overline{N}$.

Подумайте, как организовать количественную проверку и этого обстоятельства, причем здесь также интересно проследить за ходом дисперсии при очень большом случайном числе (более 40 000). Стоит поварьировать и число первоначальных малых интервалов. При некотором минимуме терпения и наблюдательности здесь также можно наткнуться на определенные сюрпризы, достойные последующего осмысления.

Правильное заполнение рабочего интервала — вовсе не единственное требование, которому должна удовлетворять хорошая случайная последовательность. Необходимо также, чтобы последовательные случайные числа, взятые попарно, также были взаимно случайными. Не станем уточнять, как это можно строго сформулировать, а вместо этого проверим нашу функцию FRAN на такие «парные корреляции».

Разбросаем по плоскости $X$, $Y$ точки, выбирая их координаты, как последовательные пары в нашей случайной последовательности:

```
3.1. X  FT(1, 128*FRAN()+1) 128*(FRAN()+1))
3.2  G  3.1
```

Смеем вас заверить, что если в самом начале образующаяся карта «звездного неба» еще будет поражать (или радовать — это уже дело вкуса) именно своей явной хаотичностью, очень скоро в ней начнет проступать подозрительная упорядоченность, переходящая постепенно в самый строгий порядок.

Сказанное относится к ФОКАЛу, но что именно получится с БЕЙСИКом, автор еще не видел.

Вот тебе и на! Ай да случайность! Какое-то гегелевское единство с необходимостью...

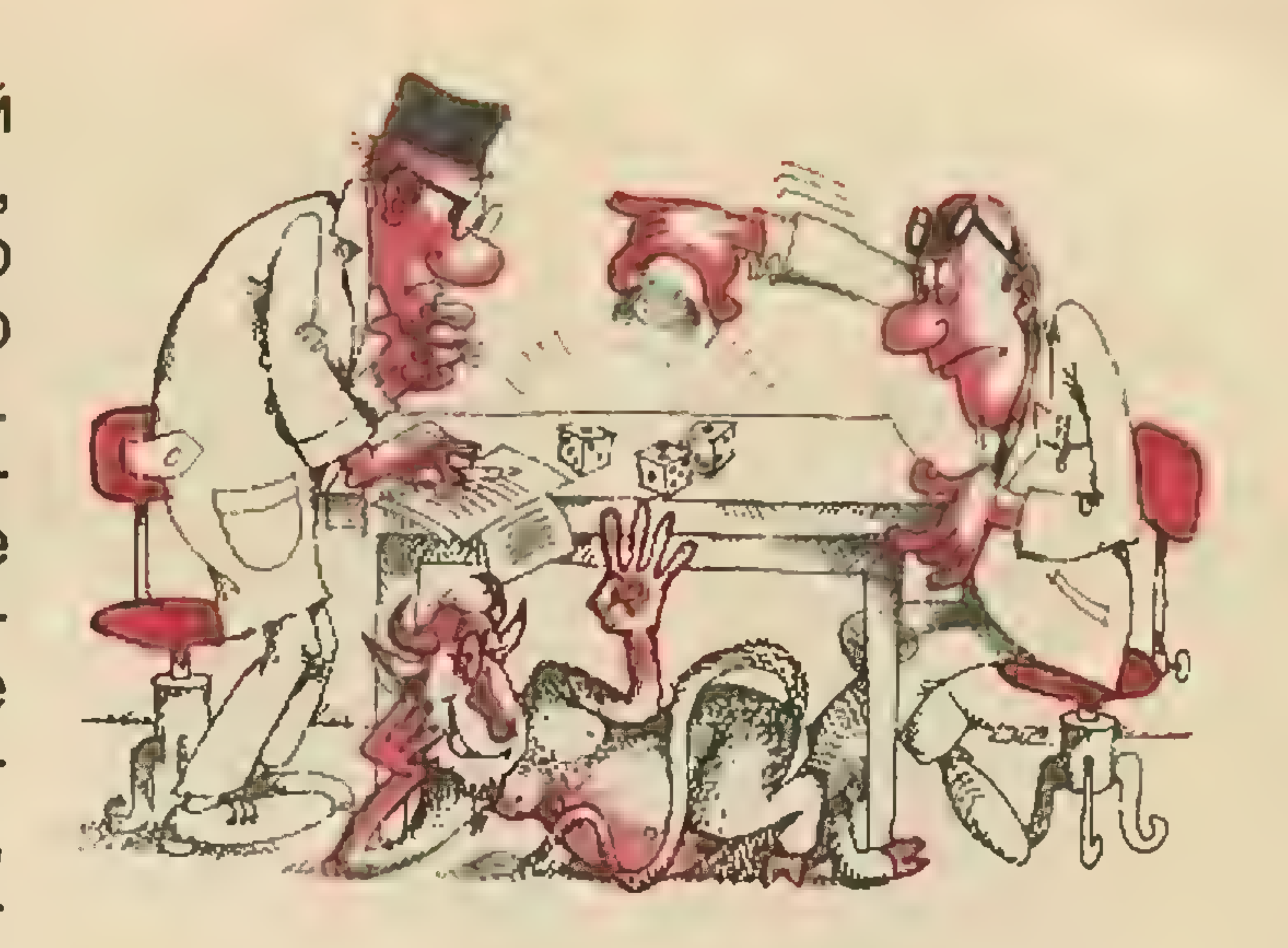
Впрочем, Гегель с его премудростью здесь ни при чем. Но может быть, дело именно в том, что мы взяли именно пары чисел и набросали их на плоскость? Снова бы попытаться окунуться в многомерность... Как ее сейчас смоделировать? Полезно опять представить, что и здесь каждое (даже холостое) обращение к FRAN эквивалентно экскурсу в «невидимое» измерение. Итак, приглашаем читателя, вставляя между нашими отстраиваемыми на дисплее парами координат несколько холостых FRAN в разных сочетаниях, попытаться добиться максимума беспорядка.

Упорядоченность в трехмерном пространстве можно наглядно выявить, если организовать косоугольную (а можно и перспективную) проекцию трехмерного пространства на экран. Это полезное упражнение может понадобиться по огромному числу поводов! Советуем исследовать «случайные небеса» с большим разрешением (увеличением «телескопа»).

Думаем, что тот или иной характер упорядоченности довольно скоро проступит, что охладит и эти «многомерные» ожидания. Итак, наша «случайность» упорядочена?

Присмотримся же еще раз к нашей исходной броуновской траектории, получаемой при очень большом (около 30 000) числе шагов. Не знаю, что строит при этом ваш БК, а мой всегда, т. е. независимо от того, где начнется «рисование», изображает одну и ту же достаточно диковинную фигуру, напоминающую летящее двукрылое (мы уже намекали на определенную симметрию) существо с когтистыми лапами, но без... головы! Мы предложили назвать это дикое создание фантазии БК «демоном». Видимо большинство читательских БК также изображают этого самого демона — есть основания думать, что по крайней мере в определенной серии машин все случайные последовательности просто совпадают. Но ведь в руководстве пользователя написано, что «FRAN... выдает... непредсказуемую последовательность случайных чисел!» Впрочем, упрекнуть создателей БК и его программного обеспечения нам здесь не в чем. Разве что авторы руководства, чуть-чуть увлекшись достоинствами этого произведения человеческого гения (мы имеем в виду именно БК), выразились не совсем точно. Но такое вполне простительно...

Если характеризовать существо программных принципов, на которых основана работа компьютеров, то сделать это лучше одного из наших крупнейших специалистов — академика С. В. Лебедева, по мнению автора, трудно. На лекции, прочитанной в МФТИ на заре компьютеризации в 1954 г., на которой посчастливилось присутствовать и автору, академик на вопрос: «Может ли ЭВМ думать?», ответил твердым «нет» и пояснил: «Машина... ведь она работает, как дура!» При этом имелась в виду сохраняющаяся по сию пору железная детерминированность программ работы ЭВМ. Для генерации случайных чисел компьютер в принципе может использовать и не программные средства, но так называемые физические задатчики (радиотехнический вариант, см. «Радио», № 3, 1987, с. 22). Для таких устройств проблемы обеспечения «правильной случайности», быстродействия и т. п. отнюдь не просты. Сейчас миниатюризация элементов ЭВМ довела их до весьма критической черты: в них возникли «истинно случайные» сбои (наводки, поломки и отказы — не в счет), вызываемые перезарядкой измельчавших конденсаторов, хранящих биты информации, под действием радиоактивных распадов и космических частиц. Не станем углубляться в обсуждение вопроса, не ускорят ли этот шум и иные случайности дорогу к столь желанному для некоторых искусственному разуму. Подчеркнем только, что упомянутое гегелевское соответствие случайного



и необходимого приобретает в этой связи и для компьютеров нетривиальный — квантовомеханический(!) оттенок...

Вообще оказывается, что любой компьютер, пусть самый наимощнейший, если только он работает на тех же физических и программных принципах, что и БК (а пока почти все компьютеры так и работают), даже в принципе не может выдавать непредсказуемую последовательность случайных чисел. Компьютер не может даже выдать вот такую простую и вовсе не случайную последовательность: 10 100 1000 10000 100000 1000000 1000000001 и т. д.

Оба замысловатых с виду утверждения немедленно следуют из простого положения: число состояний устройства из конечного числа элементов, каждый из которых может находиться в конечном числе состояний, также конечно.

Что же касается случайных последовательностей, то рано или поздно любая компьютерная последовательность начинает повторяться. Обычно в качестве самого первого (но далеко не исчерпывающего) критерия качества таких последовательностей указывается именно значение периода, т. е. полного количества чисел, после выдачи которого последовательность начинает строго повторяться. Предлагаем нашим пользователям самим найти значение периода FRAN. Мы уже намекали, что оно должно составлять что-то около нескольких десятков тысяч.

Для дальнейшего нам может понадобиться совершенно точное значение периода.

Если же кому-либо покажется, что найденный период слишком мал, отметим, что в БК действует 16-разрядный микропроцессор, а $2^{16}$ =65536, и это

есть вообще максимальное количество различных целых чисел, которые способны в нем возникать. Поэтому не так уж плохо иметь десятки тысяч различных случайных чисел, да и немудрено, что они рано или поздно начинают проявлять какую-то упорядоченность.

Все же отметить, что наша FRAN начинает когда-то повторяться, значит сказать еще далеко не все об ее хитросплетениях. Другая особенность, которую нелишне отметить, состоит в том, что демон — фигура замкнутая, а уж если быть вполне точным — почти замкнутая.

Поэтому при проходах нескольких периодов FRAN броуновской частицей демон почти строго начинает «повторяться», но все же не точно, а с очень малым сдвигом (при паре проходов этот сдвиг можно и не засечь). Предлагаем самим придумать тест для определения точности, с которой последовательные демоны ложатся друг на друга.

Интересно, что на моем БК результат был разным в зависимости от того, производилось ли суммирование всей последовательности или вычитание (набрать на БК минус не в пример легче, чем плюс, который надо давить одновременно с НР, причем именно давить). Исследование выявило «странно-случайное» число, номер которого, считая от начала FRAN (устанавливающегося после включения в сеть), равен 51. Хотя оно равно вроде бы −1, если его вычесть из самого себя, получится не нуль, а −2. Подозрения в нечистоплотности моего БК рассеялись после обследования другого экземпляра. А что происходит с вашим № 51?

Отмеченные здесь особенности функции FRAN приведут к появлению ряда своеобразных отклонений от теоретически ожидаемых статистических законов, в частности и в основном законе блужданий. Возврат траектории в исходную точку (на плоскости) проявится в неоправданном статистически «занулении» зависимости $\overline{\Delta r^2(t)}$, которая, как мы отметили, может быть основательно приглажена. Не вполне же точное возвращение приведет к тому, что при большом количестве повторных проходов (строго говоря, не следует вообще пользоваться случайной функцией, обращаясь к ней большее число раз, чем значение ее периода) пропорционально времени начнет расти уже не квадрат удаления, а сама первая степень расстояния. В пространствах большей размерности, иначе в случаях, когда между двумя фиксируемыми шагами частицы обращениями к FRAN вставляется несколько холостых (или, если эта функция используется для каких-либо иных целей), траектория может и не вернуться в исходную точку, т. е. соответствующий демон не замкнется. Тогда переход к этому «неправильному» закону смещения может выявиться гораздо раньше. Уверены, что читатели, запускавшие на БК по нескольку частиц, успели заметить некоторые указанные нами особенности.

Все же, сколь нам может и не стать обидно, что коварный БК не столь хорошо соблюдает таинственные вероятностные законы, наше дело представить, где еще и каким образом это обстоятельство может сказаться в нашем дальнейшем общении с компьютером. Ну вот, например, теперь ясно, что он не может выставить в игре против нас более десятка тысяч вариантов, но для среднестатистического игрока такое количество может считаться приемлемым. Однако снова повторим, что игры — это не наша епархия... А в математике?

Оговорка о среднестатистичности сделана с тем, чтобы подчеркнуть, что люди с исключительными способностями в состоянии запомнить и не такие количества данных. Для профессиональных шулеров и любителей «очка», «железки» и других малопочтенных игр это просто вопрос квалификации. А выдающиеся шахматисты? Легенды, в изобилии ходящие о причудах чемпионов мира, утверждают, что один из них, ныне покойный, после нескольких сдач безошибочно запоминал рубашки всех карт, так что дальнейшее сражение в преферанс становилось небезопасным. Известно, что следует говорить о покойных, но думаем, что подобное исключительное качество и чемпиона характеризует положительно... Игра, «заряженная» в только что включенный в сеть БК, всегда начинается с одного и того же варианта. Полезно вводить в нее несколько предварительных обращений к FRAN, число которых определяется какими-либо случайными действиями игроков, их именами и т. п.

Мы с вами уже производили проверку FRAN на правильность заполнения квадрата. Как и для исходного отрезка, такое заполнение должно быть по крайней мере более или менее равномерным, т. е. в равные по площади части квадрата должно рассеяться в среднем одинаковое число звездочек.

Это обстоятельство часто используется для нахождения площадей различных фигур, т. е. для вычисления интегралов. Набросаем в квадрат *n* случайных звездочек, подсчитаем, сколько попало внутрь исследуемой фигуры, разделим результат на *N* и найдем грубо требуемую площадь.

Все методы, использующие законы случая, получили обобщающее название методов Монте-Карло. В ученом собрании теперь можно услышать: «Вон идет известный монтекарлист!» Название произведено от места дислокации некогда наимоднейшего казино с его рулеткой — символом случайности.

Применим этот метод к вычислению площади круга, которая, правда, теперь и без того настолько хорошо известна, что даже поэт сказал: «$\pi R^2$ есть площадь круга!» Поэты же, как все знают, никогда не отличались пристрастием к точным наукам.

И вот один поэт договорился до утверждения (теоремы): «Площадь круга, площадь круга, два пи эр!» Там, правда, упоминается некая дама из АПН, с которой поэт непрочь познакомиться... Может, это все его и сбило. Но такое неевклидово решение проблемы квадратуры заметил неутомимый А. Иванов, который не упустил и по сему поводу создать блестящий едкий опус. Эта геометрическая дискуссия лишь добавила вящей славы и без того славному поэту да и самому А. Иванову...

Но тем не менее:

обобщение нашей задачи на случай трехмерной, а затем и *n*-мерной сферы (гиперсферы). Кстати, если для шара ответ все обязаны знать, то для *n* измерений нелишне будет попытаться как-то найти результат хотя бы по справочникам, а возможно, и непосредственным расчетом.

Число лишних знаков, добытых в числе Архимеда (термин из руководства пользователя), давно стало символом производительности ЭВМ и мастерства программистов. Хотя математика учит, что в записи этого числа в силу его трансцендентности можно в скрытом виде найти все мыслимые сведения и законы, мы не станем тягаться с профессионалами. Мой БК, прожевав всю FRAN, не добыл методом Монте-Карло и трех правильных знаков. Как ваши БК? (См. также задачу в конце этой статьи). Дальнейшие обобщения: найти моменты инерции фигур, тел, гипертел...

Кстати, если уж затронут вопрос о производительности, его пора начать исследовать экспериментально. Скажем, сколько времени нужно БК, чтобы выдать очередное случайное число? Чтобы ответить на этот вопрос, придется засечь время, необходимое для совершения примерно 1000 однородных операций (организовав соответствующий цикл), из которых в одном случае все содержат обращение к FRAN, а в другом — нет. Таким путем можно оценить, что необходимо примерно 5 мс. Этот результат следует всегда иметь в

```
4.1  I ((FRAN()) ˥2+(FRAN()) ˥2) — 1) 4.3
4.2  SA=A+1; SP=4*B/A; X FK(0,0); TP, A; G 4.1
4.3  SB=B+1; G 4.2
```

Как и в большинстве чисто вычислительных задач, нас в этом примере обязан волновать вопрос о производительности ЭВМ, т. е. о времени, необходимом для вычисления числа пи с определенной точностью. Постарайтесь определить, в состоянии ли БК, пользуясь предложенной простой программой, найти пи с точностью лучшей, чем в старинном приближении 22/7? Можно ли добиться существенного улучшения точности и какие меры для этого понадобятся? А как с затратами времени?

Для дальнейшего знакомства с возможностями вероятностных методов вычислений на БК можем также предложить естественно напрашивающееся

виду, в частности, при любых попытках «улучшить» FRAN, дополнительно его перетасовать, создать конкурирующую программу или даже внешний «физический задатчик». Полезно аналогичные исследования провести и для других функций, которыми владеет БК, — FSIN, FEXP и т. д.

Однако не станем слишком отвлекаться от намеченного генерального направления и вернемся к нашей броуновской частице. Добавим к смещению по координате *Y* (строка 1.3) на каждом шагу небольшую постоянную добавку (0.01, 0,1, 0,5). Запустите видоизмененную программу. Что за процесс она теперь моделирует? Измените знак

добавки. Какой объект в природе может изображаться теперь нашей точкой? Устроим из верхней границы экрана непроницаемую (отражающую) преграду и понаблюдаем за нашей частицей некоторое время. Что же это все-таки за частица?

Постоянное на каждом шагу смещение моделирует движение с постоянной скоростью, которая характерна для движения в вязкой среде под действием постоянной силы (в отличие от постоянного ускорения для свободной частицы). В зависимости от соотношения величины этой добавки со средним броуновским шагом характер результирующей траектории может заметно различаться. Напомним, что в молекулярной физике движение в вязкой среде под действием внешней силы на фоне случайных блужданий называется дрейфом. Если частица дрейфует в какой-либо жидкости вниз под действием силы тяжести, соответствующий процесс называется осаждением, седиментацией (это одно и то же) или даже отмучиванием. Всплывающая же вверх под действием архимедовой силы (она несколько больше веса) капелька жира в молоке участвует в процессе сепарации, приводящем к отделению сливок.

Эти процессы находят важное практическое применение всюду, где нужно что-либо разделить по плотности, размерам. Сепарация молока — серьезнейшее дело, для него приспосабливалась и центрифуга, и паровая турбина. Фирма, основанная некогда Лавалем, продолжает совершенствовать и это «сливочное» направление, выпускает массу и других быстровращающихся изделий до газотурбинных двигателей и т. п.

Явление дрейфового движения характерно и для некоторых других физических систем. Так, электроны, движущиеся в обычных проводниках, перенося столь важные для нас токи, также дрейфуют, причем для них броуновская, хаотическая составляющая движения гораздо сильнее выражена, нежели направленная. Легко прикинуть, что средние дрейфовые скорости электронов в обычной меди не превышают 1 мм/с, тогда как тепловые скорости порядка $10^6$ м/с. Пусть читатель припомнит какие-либо иные примеры...

На этом мы можем пока прервать наше предварительное знакомство со сферой случайного с помощью БК. В дальнейшем мы еще не раз будем обращаться к нашей FRAN и как чистые потребители, и как исследователи, поскольку нас не может не интересовать сам процесс производства случайных чисел. Для того чтобы читатель мог хотя бы в самых общих чертах представить, насколько это в целом дело непростое, мы можем порекомендовать ему самым поверхностным образом пробежать взглядом профессиональное исследование (см.: Добрис Г. В. Генерация псевдослучайных чисел и статистические испытания // Программирование. — 1986. — № 2).

Наконец, порекомендуем еще пару тем, связанных с исследованием случайного, которым и читатель также смог бы заняться самостоятельно (обе навеяны перелистыванием раздела «Занимательный компьютер» из журнала «Жизнь науки»).

1. На дощатый пол беспорядочно высыпается множество спичек. Ширина доски равна двум длинам спички. Какой будет доля спичек, пересекающих щели между досками (задача Бюффона, который говорил, правда, об иголках)? Не станем подсказывать ожидаемый теоретический результат. Посмотрим, что легче окажется читателю: найти этот результат именно теоретически или «в машинном эксперименте» на БК.

2. Постройте на БК модель известной доски Гальтона, представляющей набор рядов гвоздиков или колышков, на каждом из которых падающий сверху шарик может с равной вероятностью отклониться либо влево, либо вправо. Не будем пока приводить других подробностей устройства доски, предоставив простор конструкторской мысли наших пользователей. Результат каждого столкновения БК может найти просто по знаку очередного случайного числа. Заметим, что распределение шариков по ячейкам доски теоретически пропорционально биномиальным коэффициентам и при большом числе рядов оно стремится к функции Гаусса (это, собственно, и называется законом больших чисел). Предлагаем подумать, как наилучшим образом организовать сравнение результатов с этой функцией.

Приятных случайных блужданий!



**Эта статья предназначена тем, кто уже изучил операторы БЕЙСИКа, однако еще не успел приобрести достаточно опыта в написании программ. Приводимые здесь рекомендации и приемы сохранят свое значение для многих ПЭВМ и диалектов БЕЙСИКа.**

**Б. Б. МАТВЕЕВ**

# СПАГЕТТИ В КОМПЬЮТЕРИЗАЦИИ

Обычно, прочтя за короткое время пособие по БЕЙСИКу, человек с техническим образованием восклицает: «Как все просто и понятно!» Действительно, персональные ЭВМ и средства работы с ними с тем и создавались, чтобы их освоение не отнимало много времени от основных занятий владельца, кем бы он ни был по профессии — врачом или строителем, физиком или бухгалтером. Более того, если для данной ПЭВМ в изобилии имеются все необходимые программы, обеспечивающие потребности потенциального пользователя, то ему зачастую вовсе нет необходимости что-либо знать о БЕЙСИКе и программировании. Достаточно лишь освоить правила общения с конкретной, уже имеющейся в продаже программой.

Однако зачастую пользователь имеет не столь благоприятную ситуацию, и объективная необходимость сталкивает его с более или менее значительным объемом программной работы.

Когда же программа начинает разбухать прямо-таки на глазах, когда пользователь начинает систематически сталкиваться со множеством ошибок, с непредсказуемым поведением программы в очевидных, казалось бы, ситуациях, прозвучавшее в начале работы «как просто» сменяется недоумением и досадой. Действительно, для создания сколько-нибудь серьезных программ мало освоить операторы, составляющие язык. Необходимо еще научиться грамотно их применять. Ситуация совершенно аналогична изучению иностранных языков, когда освоение словарного минимума и простых диалогов отнюдь не решает задачи полноценного общения. Итак, как знает любой программист, от формального знакомства с алгоритмическим языком до серьезного умения работать на нем дистанция огромного размера. Постараемся показать это читателю на конкретных примерах.

В ходе дальнейшего изложения мы приведем некоторые иллюстрации работы с БЕЙСИКом на примере построения простой базы данных, приспособленной под хранение, например, библиографических ссылок. Разумеется, пользователь, не интересующийся таким применением программы, сможет легко переоборудовать ее под другие цели.

Начнем с простого примера.

Пусть нам нужно организовать ввод фамилий авторов ссылок. Осуществим это с помощью следующей программы:

```
10  REM                    ПРИМЕР 1.
15  REM               ВВОД ФАМИЛИЙ АВТОРОВ
20  REM                 18 ФЕВРАЛЯ 1988 Г.
100 DIN FAM¤(100)
110 REM ###### УЧАСТОК ВВОДА ######
115 K1=1
120 PRINT " НАЧИНАЕМ ВВОД ССЫЛОК"
122 PRINT " "
130 REM---
140 INPUT"ФАМИЛИЯ:";NAME¤
150 IF NAME¤="****" THEN GOTO 190
160 FAM¤(K1)=NAMF¤
170 K1=K1+1
180 GOTO 130
190 REM+++
195 NMAX=K1
200 PRINT "ВЫ ВВЕЛИ СЛЕДУЮЩИЙ СПИСОК ФАМИЛИЙ"
205 NMIN=NMAX-1
210 FOR K1=1 TO NMIN STEP 1
220 PRINT FAM¤(K1)
230 NEXT K1
240 PRINT " "
250 PRINT " СПИСОК ФАМИЛИЙ ВЫВЕДЕН."
```

Программа как таковая начинается здесь со строки 100, которой предшествует заголовок. Необходимо выработать привычку делать хотя бы небольшой, в две-три строки, комментарий в начале программы, чтобы, имея быстро растущее программное хозяйство, не тратить время, пытаясь по виду текста вспомнить его содержание. Можно смело утверждать, что при отсутствии комментариев, вернувшись к знакомой программе через пару недель, даже сам ее автор только с усилием вспомнит, который именно из десятка вариантов текста он видит перед собой. Так что условимся резервировать номера строк, скажем, до 99 под заголовок программы и описания различных программных объектов.

В строке 120 выводится на экран начальная поясняющая надпись. Можно обойтись и без этого, но хорошим стилем, безусловно, будет, если программа выдает ясный и исчерпывающий набор сообщений. Оператор PRINT в следующей строке позволит сделать пропуск строки после надписи, чтобы улучшить восприятие текста на экране. Возможностью улучшить «читабельность» сообщений программы на экране не стоит пренебрегать. В строке 140 с клавиатуры вводится очередная фамилия и помещается в строковую переменную NAME¤. Фамилий вводится несколько, и ЭВМ надо объяснить, когда же этот ввод будет закончен. Можно предварительно информировать ЭВМ о количестве вводимых фамилий и оформить этот процесс циклом FOR—NEXT, задав в нем необходимое число повторений. Однако если в процессе ввода пользователь решил ввести больше фамилий, чем предполагал заранее, или же решил вовсе прекратить ввод, то возникает проблема выхода из цикла до выполнения заданного числа повторений. Поэтому изберем вариант выхода по признаку окончания. Пусть таковым будут четыре звездочки, хотя можно избрать и какой-либо другой признак, лишь бы он содержал больше одной литеры (во избежание случайного ввода) и состоял из малоупотребительных сочетаний символов.

Строка 150 сличает введенную с клавиатуры информацию и в случае совпадения с признаком окончания передает управление за пределы участка ввода. Если введенная строка не совпадает с признаком окончания, то можно запомнить ее в одном из элементов массива FAM¤, на который указывает переменная K1, начальное значение которой определено в строке 100. Запись в очередной элемент массива осуществляется в строке 160, после чего переменная NAME¤ готова для принятия нового элемента ввода, только надо не забыть увеличить на единицу указатель K1, подготовив тем самым запись в очередной свободный элемент массива FAM¤.

Участок ввода завершается строкой 180, передающей управление строке 130, очередному оператору ввода. Таким образом, программа будет располагать в массиве FAM¤ все новые фамилии, пока мы не введем признак окончания.

Обратим внимание на строку 180. Употребление оператора передачи управления GOTO является, наверное, одним из наиболее остро дискутируемых приемов программирования. И это вполне понятно. Множество переходов внутри программы способно совершенно запутать ее смысл, породить трудные для обнаружения ошибки в логике ее работы. При этом наиболее неприятными являются передачи управления «назад» по тексту программы. В нашем случае употребление GOTO оправдано тем, что участок ввода, по-видимому, не будет сильно разрастаться в длину, а логика его работы ясно понимается из его текста. Общей рекомендацией может быть очень умеренное употребление GOTO, особенно идущих «назад», в частности, преимущественное употребление цикла FOR—NEXT с большим заданным числом повторений и выходом по условию примерно так, как это сделано в строке 150.

Строка 130. Управление на нее передается «назад» от строки 180 оператором GOTO. Почему же переход осуществляется к комментарию, а не прямо к строке 140? Дело в том, что

текст программы находится в процессе постоянного изменения. Допустим, вы захотите добавить какие-то операторы в участок ввода до оператора INPUT. Тогда, чтобы они находились внутри этого участка и использовались при каждом вводе, нам нужно будет при каждом таком добавлении поправлять строку 180, меняя адрес перехода на номер добавляемого оператора. Это чревато множеством недоразумений и лишней работы. Достаточно вспомнить, что вероятность внести ошибку при одном исправлении составляет 20% [1]. Лучше дать постоянную ссылку на строку с комментарием, пометив его, скажем, тремя минусами, что будет указывать на «неправильную» передачу управления «снизу». Теперь явно видно, что на строку 130 идет откуда-то снизу передача управления, и ее нельзя стирать или произвольно изменять ее номер. При необходимости после нее можно добавлять в участок ввода другие строки и модифицировать их, менять их номера. Аналогичную роль играет строка 190. Она отмечена тремя плюсами как «правильная» передача управления «сверху вниз». Строка 195 позволяет запомнить последнее значение переменной ввода K1 в переменную NMAX для дальнейшего использования. В БЕЙСИКе все переменные «глобальны», т. е. областью их действия является вся программа, поэтому лучше искусственно «локализовать» текущие переменные в циклах и в местах, подобных рассмотренному участку ввода. При требовании сохранения за пределами цикла каких-либо их значений будем выделять специальные переменные. В данном случае нам необходимо сохранить номер первого незанятого элемента массива FAME¤. Этот номер затем может неоднократно требоваться программе, и лучше его поместить в специальную переменную NMAX, в то время как «локальная» переменная K1 будет снова использована как переменная цикла в строках 220—240 для контрольного вывода на экран введенной перед этим информации. Настоятельно рекомендуется выводить для визуального контроля на экране введенные величины и строки. Это — необходимый прием даже при надежно работающей клавиатуре. Он называется «эхо-печать», т. е. вывод на экран информации, непосредственно перед этим введенной. Можно также распечатать введенный массив фамилий после окончания ввода, как мы и поступили. Если не принимать таких мер предосторожности, то мы не будем иметь никакой гарантии в достоверности введенной информации.

Как смог убедиться читатель из обширного комментария к такой, казалось бы, невзрачной программе, как приведенная в примере 1, создание надежной, удобной в работе и модификации, так сказать, «непотопляемой» программы является не просто применением суммы знаний о языке, а результатом определенной технологии программирования. Рядовой пользователь ПЭВМ, не располагающий ни печатающим устройством для выдачи полного листинга программы, ни какими-либо средствами отладки, может видеть на экране лишь небольшой участок текста в два десятка строк. В таких условиях факторы надежности, ясности, логичности, хорошей читаемости программы в громадной степени будут определять успех (или неуспех) всего замысла.

Конечно, пользователь, например БК-0010, привыкший тесниться в жестких пределах, отведенных ему 16 К, может охарактеризовать введение многих независимых имен переменных, дополнительные строки с комментариями и т. д. как некоторую расточительность по отношению к доступной памяти.

Но это только кажущийся проигрыш. Написание за один этап плотного, компактного, высокоэффективного кода посильно весьма немногим. Можно согласиться, что экономия памяти или эффективность во многих случаях являются необходимыми и даже решающими качествами программы. Однако все хорошо в свое время. Вот что пишет автор известной книги о технологии программирования преподаватель из Калифорнии Денни Ван Тассел [1, с. 117]: «Основной задачей программирования является создание правильных, а не эффективных программ.

Эффективная программа не нужна, если она не обеспечивает правильных результатов. Это правило Ван Тассела. Эффективная, но неправильная программа редко может быть сделана правильной, в то время как правильную, хотя и неэффективную, программу можно оптимизировать и сделать эффективной».

Приведенная выше цитата — это одна из иллюстраций существующего положения вещей: невозможно создание программы при одновременном учете многих противоречащих друг другу критериев (надежности, эффективности, структурированности, мобильности и многих других). Всегда приходится отдавать предпочтение какой-то одной группе критериев в ущерб другим. Поэтому поставим себе в качестве первоочередной цели написание прежде всего правильной, надежно работающей программы.

Несомненно, надо не упускать из виду, что рано или поздно растущая программа может исчерпать ресурсы памяти или начать работать слишком медленно. При приближении подобной ситуации будет необходимо оптимизировать участки кода, работающие неудовлетворительно.

Перейдем к следующему примеру. Конечно, для запоминания библиографической ссылки мало одной фамилии. Будем также запоминать и все остальные необходимые данные: страницу, том, заголовок работы и т. п. Кроме этого, организуем поиск ссылки по фамилии автора. Дополнив текст примера 1, получим программу для ссылок:

```
10  REM                  ПРИМЕР 2.
15  REM         ВВОД БИБЛИОГРАФИЧЕСКИХ ССЫЛОК
20  REM              18 ФЕВРАЛЯ 1988 Г.
100 DIM FAM¤(100)
101 DIM TITLE¤(100)
102 DIM VOL¤(100)
103 DIM NUM¤(100)
104 DIM PAGE¤(100)
105 DIM YEAR¤(100)
106 DIM PUBL¤(100)
107 REM---
110 REM ###### УЧАСТОК ВВОДА ######
115 K1=1
120 PRINT " НАЧИНАЕМ ВВОД ССЫЛОК"
122 PRINT " "
130 REM---
140 INPUT "ФАМИЛИЯ:";NAME¤
150 IF NAME¤="****" THEN GOTO 190
151 INPUT "ЗАГЛАВИЕ:";TIT¤
152 INPUT "ТОМ:";VOLM¤
153 INPUT "НОМЕР:";NUMER¤
154 INPUT "СТРАНИЦА:";PAG¤
155 INPUT "ГОД ИЗДАНИЯ:";YE¤
156 INPUT "ИЗДАТЕЛЬСТВО:";PUB¤
160 FAM¤(K1)=NAME¤
161 TITLE¤(K1)=TIT¤
162 VOL¤(K1)=VOLM¤
163 NUM¤(K1)=NUMER¤
164 PAGE¤(K1)=PAG¤
165 YEAR¤(K1)=YE¤
166 PUBL¤(K1)=PUB¤
170 K1=K1+1
180 GOTO 130
190 REM+++
195 NMAX=K1
200 PRINT "ВЫ ВВЕЛИ СЛЕДУЮШИЯ СПИСОК ФАМИЛИЯ:"
205 NMIN=NMAX-1
210 FOR K2=1 TO NMIN STEP 1
220 PRINT FAM¤(K2)
230 NEXT K2
235 PRINT " "
240 PRINT " СПИСОК ФАМИЛИЙ ВЫВЕДЕН."
250 REM
```

```
260 REM ###### УЧАСТОК ПЕРЕКЛЮЧЕНИЯ РЕЖИМОВ ######
270 REM
280 PRINT " МЕНЮ РЕЖИМОВ РАБОТЫ          "
290 PRINT " "
300 PRINT " 1. ПОИСК  ФАМИЛИИ  АВТОРА    "
310 PRINT " 2. ВВОД      БИБЛИОГРАФИИ    "
330 PRINT " "
340 REM---
350 IF INKEY¤="1" THEN GOTO 400
360 IF INKEY¤="2" THEN GOTO 130
370 GOTO 340
400 REM+++
401 INPUT "ВВЕДИТЕ ФАМИЛИЮ ДЛЯ ПОИСКА:";NAME2¤
402 PRINT "ФАМИЛИЯ ДЛЯ ПОИСКА: ";NAME2¤
403 REM---
404 INPUT "НАЧАТЬ ПОИСК? (Y/N) "; NAME3¤
405 IF NAME3¤="Y" THEN GOTO 410
406 IF NAME3¤="N" THEN GOTO 250
407 PRINT "НЕВЕРНЫЙ ОТВЕТ. ПОВТОРИТЕ ВВОД."
408 GOTO 403
410 REM+++
411 FOR K3 = 1 TO NMIN
420 IF FAM¤(K3)=NAME2¤ THEN GOTO 450
430 NEXT K3
440 PRINT "ФАМИЛИЯ ";NAME2¤;" НЕ НАЙДЕНА"
445 GOTO 500
450 REM+++
451 PRINT "НАЙДЕНА ФАМИЛИЯ АВТОРА: ";FAM¤(K3)
452 PRINT " НОМЕР ССЫЛКИ В СПИСКЕ  ";K3
454 PRINT "                 ЗАГОЛОВОК: ";TITLE¤(K3)
456 PRINT "                       ТОМ: ";VOL¤(K3)
458 PRINT "                     НОМЕР: ";NUM¤(K3)
460 PRINT "                  СТРАНИЦА: ";PAGE¤(K3)
462 PRINT "                       ГОД: ";YEAR¤(K3)
464 PRINT "              ИЗДАТЕЛЬСТВО: ";PUBL¤(K3)
500 END
```

Начало примера 2 во всем аналогично предыдущему тексту, с той лишь разницей, что теперь в соответствующие массивы вводится полный набор информации о библиографической ссылке. Условный оператор в строке 150 позволяет, как и раньше, прекратить ввод при получении признака окончания в переменной NAME¤. После ввода и распечатки массива мы ввели участок «переключения» режимов. Программа, достигнув строки 340, начнет «вращаться» по участку строк 340—370, пока не будет нажата клавиша 1 или клавиша 2. Чтобы пользователь не был введен в заблуждение кажущимся бездействием программы (хотя программа работает, на экране ничего не происходит), перед таким «переключателем» обязательно должна выдаваться на экран надпись-подсказка. Когда же режимов выбора станет больше, то все их необходимо перечислить в поясняющей надписи на экране. Такой способ выбора одного из режимов работы программы называется «выбор из меню». Поскольку участок строк 340—370 будет проходиться неопределенно большое число раз, цикл здесь неприемлем, и мы допустим «неправильный» GOTO, что можно мотивировать, как и в строках 130—180, краткостью участка текста и простотой его функционального назначения.

Выбрав в меню вариант 1, мы перейдем к участку программы, оформленному в виде цикла (строки 400—450), который позволяет «пройтись» по всему введенному файлу в поисках фамилии автора, совпадающей с введенной в строке 402. Мы применили здесь немедленную эхо-печать, поскольку даже единственная неверно введенная буква в фамилии будет причиной поиска и сравнения с неверным образцом. Зарезервируем пока строку 500 как самую последнюю в программе. Переход к ней теперь будет означать выход из программы.

Рассмотрим структуру программы в целом. Теперь можно убедиться, что

стоило программе немного вырасти, как проявился целый ряд неудобств и даже ошибок. Например, ввести ссылки можно только один раз, искать фамилию тоже лишь однократно. Кроме того, к участку ввода теперь можно многократно обращаться из меню, и при каждом обращении ввод будет производиться снова и снова с первого элемента массивов, затирая тем самым прежде введенную информацию. Это — типичный пример неточного определения смысла переменных. Но лучше поздно, чем никогда. Мы ввели ранее переменную NMAX, указывающую на первый незанятый элемент массивов. Зафиксируем теперь это определение и запретим использование этого имени для других целей. Теперь это будет указатель, изменяющийся вместе с размером самого файла. Менять значение указателя можно будет только в тех участках программы, в которых будет производиться ввод или вычеркивание ссылок. Переменную NMAX теперь будем инициализировать явно в начале программы, а переменная участка ввода *N* перед каждым входом в него будет принимать значение первого свободного элемента массивов, а в конце ввода будет заносить обновленное значение указателя из *N* обратно в NMAX.

Однако нами еще отнюдь не решены проблемы общего построения программы, ее структуры. Теперь хорошо видно, что самый естественный для пользователя порядок работы должен начинаться с выдачи на экран основного меню (строки 250—330). Пользователь выбирает из меню нужный режим, работает в этом режиме, а по окончании возвращается обратно к меню для следующего выбора. Очевидно, что с самого начала программы надо было поместить участок набора режимов, управление с которого передавалось бы к конкретным режимам. В теперешнем же виде программы после ее запуска мы оказываемся на участке ввода. Разумеется, формально это легко исправить, вставив строку 115 — GOTO 250, а в комментарии строки 250 добавить «+++» как указание на ссылку «сверху». Однако как метод такое «развязывание» нежелательно, ибо, будучи постоянно применяемо, приводит к появлению множества переходов вверх и вниз по тексту программы.

Такой способ решения проблем носит название «спагетти», поскольку выявить логику участка программы, запутанного множеством переходов, или модифицировать такой текст столь же трудно, как отделить от остальных одну из макаронин, перепутанных в тарелке. Поневоле допустив появление одной такой макаронины (GOTO 250), впредь постараемся быть предусмотрительнее. Ревнителям же абсолютной чистоты и структурности текста остается только ввести текст программы заново, изменив порядок операторов и аккуратно исправив номера строк в операторах перехода.

Из всего сказанного выше можно сделать вполне естественный вывод: если бы мы детально представили себе с самого начала работу программы и сформулировали ее желаемые функции, переделок удалось бы избежать. Нарисуем теперь блок-схему структуры программы (рис. 1). Каждый из прямоугольников означает участок кода, выполняющий определенную функцию.

Мы предусмотрительно зарезервировали пустые прямоугольники под функции, которые могут появиться в дальнейшем. Из получившейся схемы явствует, что каждую из стрелок-связей на рис. 1 программно надо оформить с помощью оператора GOSUB. Пунктирные стрелки от нижнего ряда прямоугольников означают дальнейшие вызовы подпрограмм. В самом верху располагается программа-диспетчер, головная программа, вызывающая отдельные операции. Такой

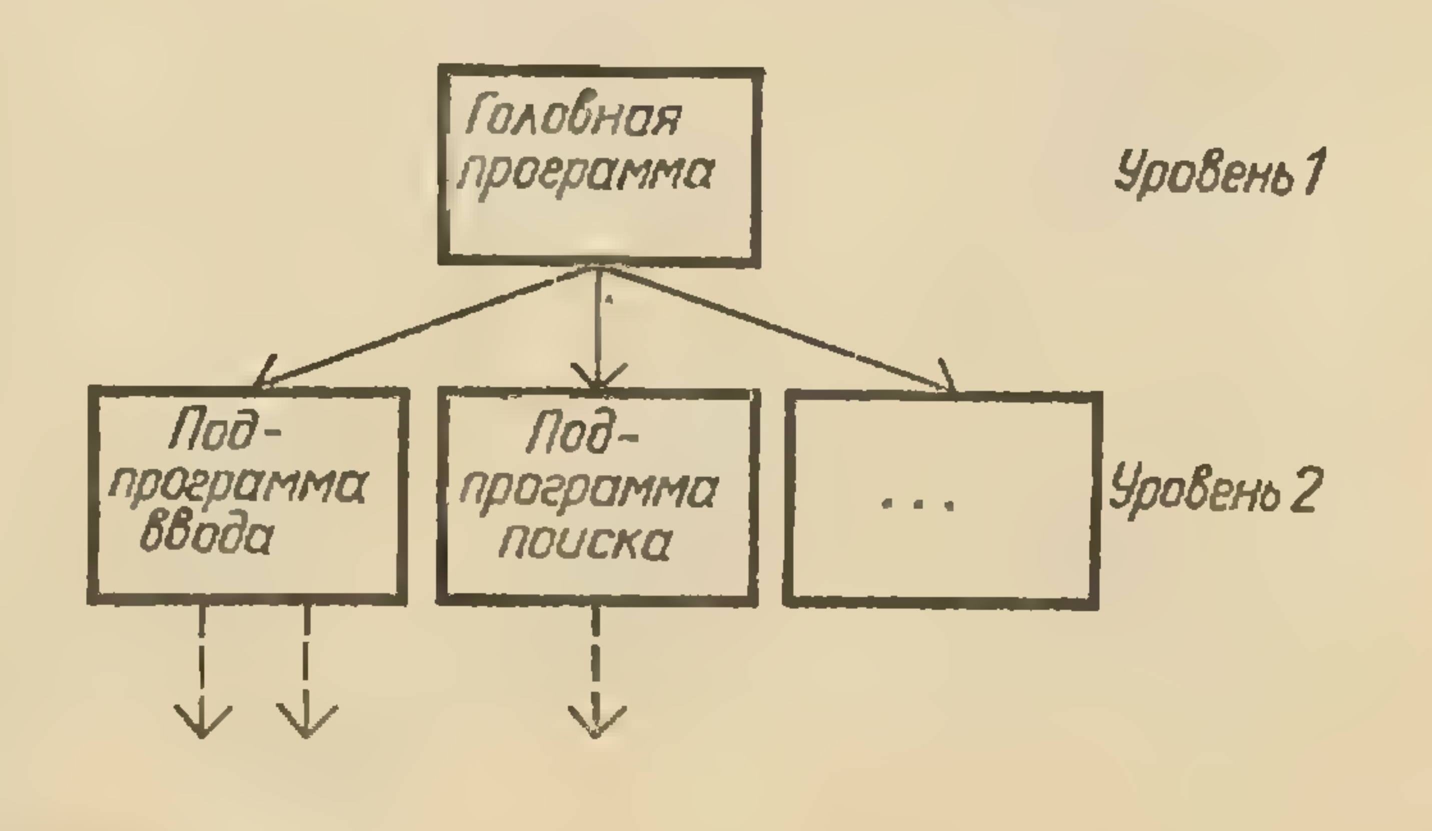



Рис.1.

головной модуль обязательно существует при правильной организации программы. Он не должен быть большим: его основная функция — дать как бы план или конспект всей программы, с необходимыми комментариями и описаниями переменных и массивов в начале и последующими вызовами тех или иных подпрограмм.

Каждая из подпрограмм тоже выполняет определенную функцию, смысл которой может быть сформулирован одной короткой фразой. Если формулировка получается пространная, то необходимо разделить подпрограмму на несколько подпрограмм с более узким функциональным назначением. Текст одной подпрограммы должен содержать не более 60 строк (примерно 1 страница текста) [2] и должен быть построен так, чтобы за 10 мин знакомства с ним можно будет понять не менее 90% его содержания (правило 90/10) [1].

Условимся также и об именах переменных. Пусть, как и в примерах 1 и 2, имена типа K1, K2 и т. д. служат переменными циклов. Пока мы отнюдь не исчерпали ресурсов оперативной памяти, введем для разных циклов разные имена их переменных. Если необходимо запомнить отдельные значения этих переменных за пределами их циклов, то будем пользоваться для этой цели специально выделенными «глобальными» переменными (NMAX в примерах 1 и 2), инициализируя и комментируя их в начале программы. Так мы начали систематически применять приемы структурного программирования. Разумеется, принятые в этой статье соглашения могут выглядеть несколько иначе, в зависимости от привычек и опыта конкретного программиста. При этом несовпадающие наборы рекомендаций будут преследовать одни и те же цели.

Рассмотрим следующий пример:

```
10 REM                  ПРИМЕР 3.
15 REM        СТРУКТУРИРОВАННАЯ ПРОГРАММА
20 REM             18 ФЕВРАЛЯ 1988 Г.
21 REM
22 REM NMAX - УКАЗАТЕЛЬ НА ПЕРВЫЙ СВОБОДНЫЙ ЭЛЕМЕНТ ФАЙЛА
23 NMAX = 1
24 REM NMIN - УКАЗАТЕЛЬ НА ПОСЛЕДНИЙ ЗАНЯТЫЙ ЭЛЕМЕНТ ФАЙЛА
27 NMIN = 0
100 DIM FAM¤(100)
101 DIM TITLE¤(100)
102 DIM VOL¤(100)
103 DIM NUM¤(100)
104 DIM PAGE¤(100)
105 DIM YEAR¤(100)
106 DIM PUBL¤(100)
115 GOTO 250
120 REM///
121 REM ===ПОДПРОГРАММА ВВОДА ССЫЛОК===
122 K1=NMAX
123 PRINT " НАЧИНАЕМ ВВОД ССЫЛОК "
124 PRINT " "
130 REM---
140 INPUT "ФАМИЛИЯ:";NAME¤
150 IF NAME¤="****" THEN GOTO 190
151 INPUT "ЗАГЛАВИЕ:";TIT¤
152 INPUT "ТОМ:";VOLM¤
153 INPUT "НОМЕР:";NUMER¤
154 INPUT "СТРАНИЦА:";PAG¤
155 INPUT "ГОД ИЗДАНИЯ:";YE¤
156 INPUT "ИЗДАТЕЛЬСТВО:";PUB¤
160 FAM¤(K1)=NAME¤
161 TITLF¤(K1)=TIT¤
162 VOL¤(K1)=VOLM¤
163 NUM¤(K1)=NUMER¤
164 PAGE¤(K1)=PAG¤
165 YEAR¤(K1)=YE¤
166 PUBL¤(K1)=PUB¤
```

```
170 K1=K1+1
171 IF K1>=101 THEN GOTO 181
180 GOTO 130
181 REM+++
182 PRINT "ДОСТИГНУТ ПРЕДЕЛЬНЫЙ РАЗМЕР ФАЙЛА!"
190 REM+++
195 NMAX=K1
205 NMIN=NMAX-1
210 RETURN
250 REM+++
260 REM ###### УЧАСТОК ПЕРЕКЛЮЧЕНИЯ РЕЖИМОВ ######
270 REM
280 PRINT " МЕНЮ РЕЖИМОВ РАБОТЫ:              "
290 PRINT " "
300 PRINT " 1. ПОИСК  ФАМИЛИЯ  АВТОРА        "
310 PRINT " 2. ВВОД     БИБЛИОГРАФИИ         "
311 PRINT " 3. ПРОСМОТР БИБЛИОГРАФИИ         "
312 PRINT " 4. ВЫХОД ИЗ ПРОГРАММЫ            "
313 PRINT " 5. ЗАГРУЗКА ФАЙЛА С ДИСКА        "
314 PRINT " 6. СОХРАНЕНИЕ ФАЙЛА НА ДИСК      "
315 PRINT " 7. СОРТИРОВКА ПО ФАМИЛИЯМ        "
316 PRINT " 8. УДАЛЕНИЕ ЗАПИСЕЙ              "
317 PRINT " 9. КОРРЕКЦИЯ ЗАПИСЕЙ             "
330 PRINT " "
340 REM---
350 IF INKEY¤="1" THEN GOSUB 400
351 IF INKEY¤="2" THEN GOSUB 120
352 IF INKEY¤="3" THEN GOSUB 600
353 IF INKEY¤="4" THEN GOTO 9999
354 IF INKEY¤="5" THEN GOSUB 700
355 IF INKEY¤="6" THEN GOSUB 800
356 IF INKEY¤="7" THEN GOSUB 500
357 IF INKEY¤="8" THEN GOSUB 900
358 IF INKEY¤="9" THEN GOSUB 1000
370 GOTO 340
400 REM///
401 REM ===ПОДПРОГРАММА ПОИСКА ССЫЛКИ ПО ФАМИЛИИ АВТОРА===
402 INPUT "ВВЕДИТЕ ФАМИЛИЮ ДЛЯ ПОИСКА:";NAME2¤
403 PRINT "ФАМИЛИЯ ДЛЯ ПОИСКА: ";NAME2¤
404 REM---
405 INPUT "НАЧАТЬ ПОИСК? (Y/N) "; NAME3¤
406 IF NAME3¤="Y" THEN GOTO 410
407 IF NAME3¤="N" THEN GOTO 250
408 PRINT "НЕВЕРНЫЙ ОТВЕТ. ПОВТОРИТЕ ВВОД."
409 GOTO 404
410 REM+++
411 FOP K3 = 1 TO NMIN
420 IF FAM¤(K3)=NAME2¤ THEN GOTO 450
430 NEXT K3
440 PRINT "ФАМИЛИЯ ";NAME2¤;" НЕ НАЙДЕНА"
445 GOTO 470
450 REM+++
451 PRINT "НАЙДЕНА ФАМИЛИЯ АВТОРА: ";FAM¤(K3)
452 PRINT " НОМЕР ССЫЛКИ В СПИСКЕ  ";K3
454 PRINT "               ЗАГОЛОВОК: ";TITLE¤(K3)
456 PRINT "                     ТОМ: ";VOL¤(K3)
458 PRINT "                   НОМЕР: ";NUM¤(K3)
460 PRINT "                СТРАНИЦА: ";PAGE¤(K3)
462 PRINT "                     ГОД: ";YEAR¤(K3)
464 PRINT "            ИЗДАТЕЛЬСТВО: ";PUBL¤(K3)
470 RETURN
500 REM///
510 REM ===ПОДПРОГРАММА СОРТИРОВКИ ФАЙЛА БИБЛИОГРАФИИ ===
520 PRINT " ИЗВИНИТЕ, ЭТА ВОЗМОЖНОСТЬ ПОКА НЕ РЕАЛИЗОВАНА"
530 RETURN
600 REM ///
610 REM ===ПОДПРОГРАММА ПРОСМОТРА БИБЛИОГРАФИИ===
611 PRINT "В ФАЙЛЕ НАХОДИТСЯ ";NMIN;" ЗАПИСЕЙ"
612 PRINT " "
615 INPUT "ВВЕДИТЕ НАЧАЛЬНЫЙ НОМЕР ЗАПИСИ ДЛЯ ПРОСМОТРА:";I1
617 INPUT "ВВЕДИТЕ КОНЕЧНЫЙ НОМЕР ЗАПИСИ ДЛЯ ПРОСМОТРА:";I2
```

```
620 FOR K4=I1 TO I2
630 PRINT "# ";K4
640 PRINT FAM¤(K4);TITLE¤(K4);VOL¤(K4);NUM¤(K4);PAGE¤(K4);YEAR¤(K4)
650 PRINT PUBL¤(K4)
660 NEXT K4
670 RETURN
700 REM///
710 REM ===ПОДПРОГРАММА ЗАГРУЗКИ БИБЛИОГРАФИИ С ДИСКА===
720 PRINT " ИЗВИНИТЕ, ЭТА ВОЗМОЖНОСТЬ ПОКА НЕ РЕАЛИЗОВАНА"
730 RETURN
800 REM///
810 REM ===ПОДПРОГРАММА СОХРАНЕНИЯ БИБЛИОГРАФИИ НА ДИСКЕ==
820 PRINT " ИЗВИНИТЕ, ЭТА ВОЗМОЖНОСТЬ ПОКА НЕ РЕАЛИЗОВАНА"
860 RETURN
900 REM///
910 REM ===ПОДПРОГРАММА УДАЛЕНИЯ ЗАПИСЕЙ ИЗ ФАЙЛА===
920 PRINT " ИЗВИНИТЕ, ЭТА ВОЗМОЖНОСТЬ ПОКА НЕ РЕАЛИЗОВАНА"
930 RETURN
900 REM///
1000 REM ===ПОДПРОГРАММА МОДИФИКАЦИИ ЗАПИСЕЙ В ФАЙЛЕ===
1010 PRINT " ИЗВИНИТЕ, ЭТА ВОЗМОЖНОСТЬ ПОКА НЕ РЕАЛИЗОВАНА"
1020 RETURN
9999 REM+++ОКОНЧАНИЕ ПРОГРАММЫ
```

Здесь мы добавили операции ввода-вывода файла библиографии. Без записи на внешний носитель памяти созданный нами файл исчезнет при сбое питающего напряжения или при выключении ПЭВМ. Поэтому нужно периодически сбрасывать обновленный файл на магнитофон или лучше на диск (если он имеется).

Предусмотрены также операции стирания и модификации отдельных записей файла, а также полезная сортировка по фамилиям авторов.

Когда используем ПЭВМ, мы обычно стеснены как в отношении оперативной, так и в отношении внешней памяти. Поэтому будем сохранять файл всегда в отсортированном по фамилиям авторов виде, а после удаления или добавления записей будем производить «принудительную» сортировку в памяти.

В меню на строках 280—230 мы зарезервировали все упомянутые выше операции. Чтобы у нас не было ошибок из-за несуществующих пока переходов на нереализованные подпрограммы, мы оставили в местах текста, предназначенных для этих подпрограмм, их «заготовки». Такая «заготовка», называемая еще «затычкой» или «подыгрывающим модулем», хороша тем, что она дает возможность последовательно реализовывать модуль за модулем, не нарушая задуманной структуры всей программы. Незнакомому же с программой или просто забывчивому пользователю «затычки» будут выдавать вежливый, ясный ответ. Читатель, наверное, уже обратил внимание, что первая строка в подпрограммах в согласии с принятыми раньше соглашениями — комментарий. Только теперь он помечен (как начало подпрограммы) тремя символами деления (косая черта). Будем отводить для начала подпрограмм строки с номерами, кратными сотне. Также в начале каждой подпрограммы отведем строку, кратко поясняющую функцию этой подпрограммы.

Отметим, что пришло в негодность наше соглашение о номере последней строки программы. Программа растет, и нам пришлось увеличить зафиксированный ранее номер с 500 до 9999, а также отыскать и исправить указывающие на него ссылки. Это занятие, чреватое пропуском (недоисправлением) одной или нескольких ссылок, которые после удаления старой строки будут передавать управление уже несуществующей строке и прерывать задачу по ошибке. Такова цена удачных и неудачных соглашений о стандарте.

Принятый первоначально интервал между номерами строк тоже быстро исчерпал себя — кое-где номера следуют уже без перерыва. Имеются диалекты БЕЙСИКа (например, Acorn BBC), где можно использовать средство перенумерации строк (разумеется, с одновременными изменениями в соответствующих GOTO и GOSUB). В

этом случае проблем не возникает. Если же такая готовая сервисная программа или команда языка отсутствует, возникает неприятная необходимость вручную перенумеровывать участки текста и изменять ссылки на них. В таких системах лучше с самого начала нумеровать строки через значительный интервал — 20 или даже 100. Подпрограммам уровня 2 (см. рис. 1) необходимо отвести участки номеров, кратные тысяче, чтобы собрать рядом в тексте не только модули второго уровня, а и логически с ними связанные подпрограммы следующих уровней. Такое расположение не обязательно, но удобно для последующей отладки.

Чтобы предотвратить в процессе ввода-вывода превышение максимальной длины массивов (определенной в строках 100—106), в подпрограмму ввода-вывода (строки 171—182) добавлены операторы, контролирующие эту ситуацию. Максимальный размер массивов выбирается в соответствии с размером оперативной памяти ПЭВМ и потребностью пользователя.

Как хранить и обновлять файл на магнитной ленте? Прежде всего сделать несколько экземпляров. Магнитная лента имеет свойство постепенно деформироваться, так что ваша программа может однажды перестать считываться с магнитофона и будет потеряна.

Обладатель Персональной ЭВМ с дисководом может отвести для файла три определенных имени на диске.

Вид операторов ввода-вывода будет несколько различаться в зависимости от типа ПЭВМ, однако структура соответствующих подпрограмм при этом не изменится. Например, для БЕЙСИКа — Acorn BBC можно оформить ввод-вывод таким образом:

ПРИМЕР 4

```
700 REM///
710 REM ===ПОДПРОГРАММА ЧТЕНИЯ БИБЛИОГРАФИИ С ДИСКА===
720 INPUT "УКАЖИТЕ ИМЯ ФАЙЛА ДЛЯ ЧТЕНИЯ С ДИСКА";FILE¤
725 INFILE=OPENIN(FILE¤)
730 INPUT#INFILE,NMAX
735 FOR K2=1 TO NMIN
740 INPUT#INFILE;FAM¤(K2);TITLE¤(K2);VOL¤(K2);NUM¤(K2);PAGE¤(K2)
742 INPUT#INFILE;YEAR¤(K2);PUBL¤(K2)
760 NEXT K2
770 REM ПРОКОНТРОЛИРОВАТЬ РЕЗУЛЬТАТЫ ЧТЕНИЯ С ДИСКА
775 CLOSE #INFILE
780 RETURN
800 REM///
810 REM ===ПОДПРОГРАММА СОХРАНЕНИЯ БИБЛИОГРАФИИ НА ДИСКЕ===
820 INPUT " УКАЖИТЕ ИМЯ ФАЙЛА ДЛЯ СОХРАНЕНИЯ НА ДИСКЕ: ";FILE¤
825 OUTFILE = OFENOUT(FILE¤)
830 PRINT #OUTEILE,NMAX,NMIN
835 FOP K2=1 TO NMIN
840 PRINT#OUTFILE;FAM¤(K2);TITLE¤(K2);VOL¤(K2);NUM¤(K2);PAGE¤(K2)
842 PRINT#OUTFILE;YEAR¤(K2);PUBL¤(K2)
860 NEXT K2
870 REM ПРОКОНТРОЛИРОВАТЬ РЕЗУЛЬТАТЫ ЗАПИСИ НА ДИСК
875 CLOSE #OUTFILE
880 RETURN
```

В конце подпрограмм ввода-вывода оставлены строки с комментарием. В этих местах надо будет производить контроль благополучного завершения компьютером очередной операции ввода-вывода. Такая проверка может потребовать от пользователя более основательных познаний о ПЭВМ, чем те, которые нужны для понимания остального материала статьи. Поэтому оставим пока эти строки для будущего развития программы.

Рассмотрим подпрограммы, реализующие операции 7, 8, 9 основного меню. Программа была структурирована нами, и теперь можно приводить только соответствующие подпрограммы, не повторяя остального текста. За преде-

лами подпрограммы, возможно, будут добавляться только описания массивов и переменных (в отведенный участок в начале программы), а также каких-то новых функций в основном меню. Вот почему выгодно отводить участок строк в начале программы под описания, а не рассыпать определения по всему тексту. Сначала рассмотрим подпрограмму удаления ссылки:

ПРИМЕР 5.1

```
900 REM///
910 REM ===ПОДПРОГРАММА УДАЛЕНИЯ ЗАПИСИ===
915 INPUT "ВВЕДИТЕ НОМЕР УДАЛЯЕМОЙ ЗАПИСИ: ";K4
920 PRINT "НОМЕР УДАЛЯЕМОЙ ЗАПИСИ: ";K4
925 PRINT FAM¤(K4);TITLE¤(K4);VOL¤(K4);NUM¤(K4);PAGE¤(K4);YEAR¤(K4)
930 PRINT PUBL¤(K4)
935 REM---
936 INPUT " УДАЛЯТЬ? (Y/N) ";NAME¤
940 IF NAME¤="N" THEN GOTO 990
945 IF NAME¤="Y" THEN GOTO 960
950 PRINT " НЕВЕРНЫЙ ОТВЕТ! "
955 GOTO 935
960 REM+++
962 FAM¤(K4)=FAM¤(NMIN)
964 TITLE¤(K4)=TITLE¤(NMIN)
966 VOL¤(K4)=VOL¤(NMIN)
968 NUM¤(K4)=NUM¤(NMIN)
970 PAGE¤(K4)=PAGE¤(NMIN)
972 YEAR¤(K4)=YEAR¤(NMIN)
974 PUBL¤(K4)=PUBL¤(NMIN)
975 PRINT " ЗАПИСИ НОМЕР ";K4;" УДАЛЕНА"
976 NMIN=NMIN-1
978 NMAX=NMAX-1
990 REM+++
999 RETURN
```

Строки 900—999, согласно вызову из основного меню, отводятся под операцию стирания. Это — «горячая» операция, которая требует контроля в виде распечатки стираемой ссылки (920—930). Убедившись, что для удаления выбрана необходимая запись, пользователь подтверждает ее вычеркивание (935—955). Как и в операции поиска (строки 404—408), здесь воспринимаются только два ответа. При вводе чего-либо иного программа требует правильного ответа. Вычеркивание производится посредством перемещения последней записи файла на место удаляемой, чтобы в файле не образовывалось пустых мест. Вот и пригодилась переменная—указатель на последний занятый элемент массивов (NMIN). Перед завершением операции уменьшаются значения обоих указателей (NMIN и NMAX). Теперь можно выходить из подпрограммы.

Подпрограмма коррекции записи также выдает для контроля изменяемую запись (1002—1008), а затем меню для выбора конкретного элемента ссылки:

ПРИМЕР 5.2

```
1000 REM///
1001 REM ===ПОДПРОГРАММА КОРРЕКЦИИ ЗАПИСИ===
1002 INPUT "ВВЕДИТЕ НОМЕР КОРРЕКТИРУЕМОЙ ЗАПИСИ: ";K5
1004 PRINT "НОМЕР КОРРЕКТИРУЕМОЙ  ЗАПИСИ: ";K5
1006 PRINT FAM¤(K5);TITLE¤(K5);VOL¤(K5);NUM¤(K5);PAGE¤(K5);YEAR¤(K5)
1008 PRINT PUBL¤(K5)
1010 REM---
1012 PRINT " ВЫБЕРИТЕ ПОЛЕ ДЛЯ ИСПРАВЛЕНИЯ: "
1014 PRINT "             ФАМИЛИЯ АВТОРА: 1 "
1018 PRINT "                  ЗАГОЛОВОК: 2 "
1020 PRINT "                        ТОМ: 3 "
1022 PRINT "                      НОМЕР: 4 "
```

```
1024 PRINT "                    СТРАНИЦА: 5 "
1026 PRINT "                         ГОД: 6 "
1028 PRINT "                ИЗДАТЕЛЬСТВО: 7 "
1030 REM---
1032 IF INKEY¤<>"1" THEN GOTO 1036
1033 I=1
1034 NAME¤=FAM¤(K5)
1035 GOTO 1061
1036 IF INKEY¤<>"2" THEN GOTO 1040
1037 I=2
1038 NAME¤=TITLE¤(K5)
1039 GOTO 1061
1040 IF INKEY¤<>"3" THEN GOTO 1044
1041 I=3
1042 NAME¤=VOL¤(K5)
1043 GOTO 1061
1044 IF INKEY¤<>"4" THEN GOTO 1048
1045 I=4
1046 NAME¤=NUM¤(K5)
1047 GOTO 1061
1048 IF INKEY¤<>"5" THEN GOTO 1052
1049 I=5
1050 NAME¤=PAGE¤(K5)
1051 GOTO 1061
1052 IF INKEY¤<>"6" THEN GOTO 1056
1053 I=6
1054 NAME¤=YEAR¤(K5)
1055 GOTO 1061
1056 IF INKEY¤<>"7" THEN GOTO 1030
1057 I=7
1058 NAME¤=PUBL¤(K5)
1061 REM+++
1062 PRINT "ВЫ ВЫБРАЛИ ДЛЯ КОРРЕКЦИИ ПОЛЕ:  ";NAME¤
1064 INPUT "ВВЕДИТЕ ИСПРАВЛЕННОЕ ПОЛЕ: ";NAME2¤
1065 PRINT "СТАРОЕ: ";NAME¤
1066 PRINT " НОВОЕ: ";NAME2¤
1067 REM---
1068 INPUT "ЗАНОСИТЬ ИСПРАВЛЕНИЕ? (Y/N) ";ANS¤
1069 IF ANS¤="N" THEN GOTO 1082
1070 IF ANS¤="Y" THEN GOTO 1073
1071 PRINT " НЕВЕРНЫЙ ОТВЕТ! "
1072 GOTO 1067
1073 REM+++
1074 IF I=1 THEN FAM¤(K5)=NAME2¤
1075 IF I=2 THEN TITLE¤(K5)=NAME2¤
1076 IF I=3 THEN VOL¤(K5)=NAME2¤
1077 IF I=4 THEN NUM¤(K5)=NAME2¤
1078 IF I=5 THEN PAGE¤(K5)=NAME2¤
1079 IF I=6 THEN YEAR¤(K5)=NAME2¤
1080 IF I=7 THEN PUBL¤(K5)=NAME2¤
1081 PRINT "ИСПРАВЛЕНИЕ ПРОИЗВЕДЕНО"
1082 REM+++
1099 RETURN
```

Участок строк 1030—1061 заносит в переменную NAME ¤ элемент одного из массивов в зависимости от нажатой клавиши. Как и в прочих приведенных выше переключателях режимов с участием операторов INKEY ¤ программа «крутится» по строкам 1030—1061 до получения с клавиатуры одного из допустимых ответов. Строки 1062—1072 организуют, как обычно, визуальную проверку введенной информации, а в строках 1074—1080 введенная во временную переменную NAME2 ¤ информация записывается в указанный переменной I массив.

Можно сделать последние две подпрограммы гораздо компактней, удалив проверки ввода с клавиатуры, однако тогда возрастает риск введения ошибочных данных.

Нам осталось рассмотреть последнюю операцию: сортировку.

ПРИМЕР 5.3

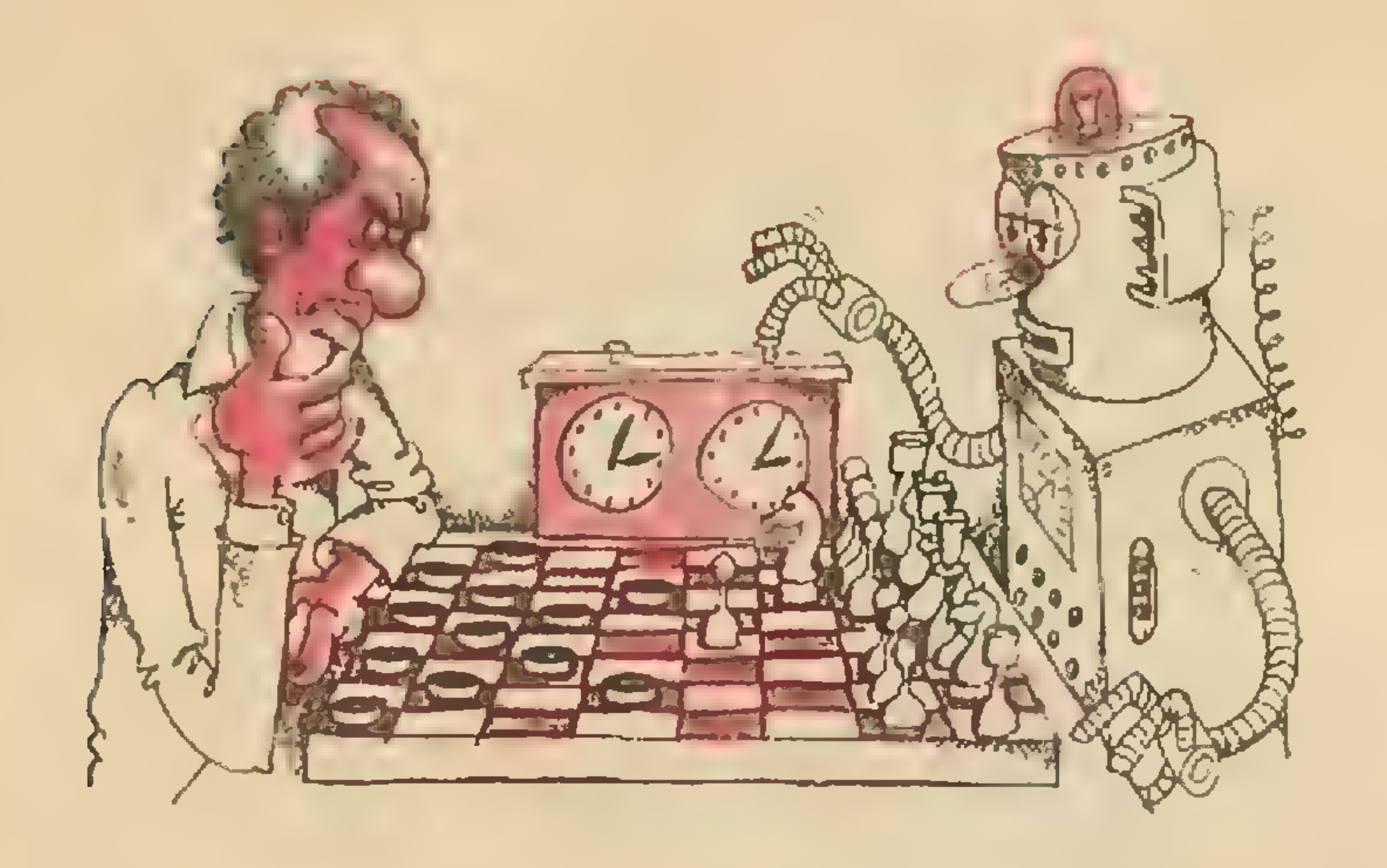
```
300 REM///
301 REM ===ПОДПРОГРАММА СОРТИРОВКИ В АЛФ. ПОРЯДКЕ ПО ФАМИЛИИ АВТОРА===
305 J=NMIN
310 REM---
311 REM  ЕСЛИ ВСЕ ОТСОРТИРОВАЛИ, ТО ИДТИ НА ВЫХОД
312 IF J<2 GOTO 342
313 REM  ПРОСМОТР ПЕРВОЙ ПАРЫ
314 I=1
315 REM---
316 REM  ЕСЛИ ПРОСМОТРЕНЫ ВСЕ ПАРЫ, ТО ИДТИ НА СЛЕДУЮЩИЙ ПРОХОД
320 IF I>J-1 GOTO 335
321 REM  ЕСЛИ ЭЛЕМЕНТЫ УЖЕ УПОРЯДОЧЕНЫ, ИХ ОБМЕН НЕ НУЖЕН
322 IF FAM¤(I)<=FAM¤(I+1) THEN GOTO 332
328 REM  ОСУЩЕСТВИТЬ ОБМЕН ЭЛЕМЕНТАМИ
330 GOSUB 370
332 REM+++
333 J=I+1
334 GOTO 315
335 REM+++
340 J=J-1
341 GOTO 310
342 REM+++
343 PRINT " СОРТИРОВКА ОКОНЧЕНА"
344 RETURN
370 REM/// ОБМЕН МЕСТАМИ ЭЛЕМЕНТОВ МАССИВОВ
371 N¤=FAM¤(I)
372 FAM¤(I)=FAM¤(I+1)
373 FAM¤(I+1)=N¤
374 N¤=TITLE¤(I)
375 TITLE¤(I)=TITLE¤(I+1)
376 TITLE¤(I+1)=N¤
377 N¤=VOL¤(I)
378 VOL¤(I)=VOL¤(I+1)
379 VOL¤(I+1)=N¤
380 N¤=NUM¤(I)
381 NUM¤(I)=NUM¤(I+1)
382 NUM¤(I+1)=N¤
383 N¤=PAGE¤(I)
384 PAGE¤(I)=PAGE¤(I+1)
385 PAGE¤(I+1)=N¤
386 N¤=YEAR¤(I)
388 YEAR¤(I)=YEAR¤(I+1)
388 YEAR¤(I+1)=N¤
389 N¤=PUBL¤(I)
390 PUBL¤(I)=PUBL¤(I+1)
391 PUBL¤(I+1)=N¤
392 RETURN
```

Здесь применена самая простая из сортировок — сортировка методом «всплывающего пузырька». Название хорошо воспроизводит порядок работы алгоритма, просматривающего элементы файла попарно и меняющего местами элементы пары, если она расположена неупорядоченно. Поэтому одни элементы сортируемого списка как бы «всплывают» к его началу, а другие спускаются ближе к его концу. Если, например, надо упорядочить по возрастанию четыре числа, то цикл по переменным I и J (строки 310—341) будет исполняться так:

| | | | | | |
|---|---|---|---|---|---|
| 10 | 10 | 10 | 10 | 10 | 1 |
| 26 | 14 | 14 | 14 | 1 | 10 |
| 14 | 26 | 1 | 1 | 14 | 14 |
| 1 | 1 | 26 | 26 | 26 | 26 |
| I=1 | I=2<br>J=4 | I=3 | I=1 | I=2<br>J=3 | I=3<br>J=2 |

Строка 330 служит вызовом подпрограммы обмена местами элементов массивов (370—392) и использует при этом временную переменную NAME¤.

К сожалению, подпрограмма сортировки, которую мы привели здесь и которую читатель может встретить почти в каждом англоязычном пособии по БЕЙСИКу, обладает существенным недостатком, а именно: она может гарантировать упорядочение только латинского алфавита. Причина заключается в следующем: действие строки 322, сравнивающей две строковые переменные, основано на побуквенном сопоставлении в сравниваемых переменных внутренних кодов ПЭВМ, кото-



рые всегда упорядочены для латинского и не всегда упорядочены для русского алфавита. Так, например, началу латинского алфавита (буквам A, B, C, D...) соответствуют коды ASCII 65, 66, 67, 68 и т. д. Тем самым будут справедливы «неравенства»: «A»< «B», «C» < «D», «BASIC» < «PASCAL», «12» < «15». Последовательность кодов для русского алфавита читатель может узнать из документации к своей ПЭВМ или же запустив такую программу:

ПРИМЕР 5.4

```
10 INPUT "ВВЕДИТЕ РУССКУЮ БУКВУ: ";A¤
15 PRINT ASC(A¤)
20 GOTO 10
```

Вводя в алфавитном порядке русские буквы, мы увидим на экране последовательность их кодов для данной ПЭВМ. Если эта последовательность не является монотонно возрастающей, придется заменить строку 332 специальной программой сравнения строк, которая и заключит собой ряд примеров БЕЙСИКа в этой статье:

ПРИМЕР 5.5

```
48 REM ОПРЕДЕЛЕНИЕ ВСПОМОГАТЕЛЬНОГО МАССИВА ДЛЯ СОРТИРОВКИ
50 DIM ALF¤(33)
52 DATA " ","А","Б","В","Г","Д","Е","Ж","З","И","Й","К","Л","М","Н"
54 DATA "О","П","Р","С","Т","У","Ф","Х","Ц","Ч","Ш","Щ","Ь","Ы","Э"
56 DATA "Ю","Я",":"
58 REM ОПРЕДЕЛЕНИЕ ПОРЯДКА БУКВ ДЛЯ СОРТИРОВКИ
60 FOR K6=1 TO 33
62 READ  ALF¤(K6)
64 NEXT K6
..................................
300 REM///
301 REM ===ПОДПРОГРАММА СОРТИРОВКИ РУССКОГО И ЛАТИНСКОГО АЛФАВИТОВ===
305 J=NMIN
310 REM---
311 REM  ЕСЛИ ВСЕ ОТСОРТИРОВАЛИ, ТО ИДТИ НА ВЫХОД
312 IF J<2 GOTO 342
313 REM ПРОСМОТР ПЕРВОЙ ПАРЫ
314 I=1
315 REM---
316 K=0
317 REM  ЕСЛИ ПРОСМОТРЕНЫ ВСЕ ПАРЫ, ТО ИДТИ НА СЛЕДУЮЩИЙ ПРОХОД
320 IF I>J-1 GOTO 335
321 REM  ЕСЛИ ЭЛЕМЕНТЫ УЖЕ УПОРЯДОЧЕНЫ, ТО I=0 И ОБМЕН ИМИ НЕ НУЖЕН
322 GOSUB 345
323 IF K=0 THEN GOTO 332
328 REM  ОСУЩЕСТВИТЬ ОБМЕН ЭЛЕМЕНТОВ:
330 GOSUB 370
332 REM+++
333 I=I+1
334 GOTO 315
335 REM+++
340 J=J-1
341 GOTO 310
342 REM+++
343 PRINT "СОРТИРОВКА ОКОНЧЕНА"
344 RETURN
345 REM/// ПОДПРОГРАММА СРАВНЕНИЯ ДВУХ ЭЛЕМЕНТОВ МАССИВА
346 K=0
347 L¤=MID¤(FAM¤(I),1,1)
348 M¤=MID¤(FAM¤(I+1),1,1)
349 IF L¤<>":" OR M¤<>":" THEN GOTO 352
```

```
350 IF FAM¤(I)>FAM¤(I+1) THEN K=1
351 GOTO 368
352 REM+++ЦИКЛ СРАВНЕНИЯ ДВУХ СТРОК
353 MAX=LEN(FAM¤(I))
354 FOR K7=1 TO MAX
355 L¤=MID¤(FAM¤(I),K7,1)
356 M¤=MID¤(FAM¤(I+1),K7,1)
357 FOR K8=1 TO 33
358 IF ALF¤(K8)=L¤ THEN IM=K8
359 IF ALF¤(K8)=M¤ THEN IP=K8
360 NEXT K8
361 IF IP>IM THEN GOTO 368
362 IF IP=IM THEN GOTO 365
363 IF IP<IM THEN K=1
364 GOTO 368
365 REM+++
366 NEXT K7
368 REM+++
369 RETURN
```

По начальным номерам строк можно видеть, что нам пришлось вставить в начало программы определение и заполнение нового массива, задающего порядок букв русского алфавита. Разумеется, в нашей власти определить и любой другой порядок следования, в зависимости от целей конкретной операции упорядочивания. Здесь мы определили двоеточие как последний символ нашей последовательности. С двоеточия будем начинать ввод всех иностранных фамилий. Таким образом, в файле будут следовать сначала русские фамилии, а затем иностранные.

В уже приводившейся программе сортировки определена новая переменная K. Это — переменная состояния сравнения, принимающая только два значения: 0 (элементы упорядочены, и менять их местами не надо) и 1 (элементы расположены с нарушением порядка, и их надо поменять местами). Переменные, определенные только для таких двух значений, обычно именуют «флагами», а относительно их состояний говорится: 0 — «флаг опущен», 1 — «флаг поднят». У нас изначально флажок опускается (строка 316) и поднимается, если в процессе сравнения двух строк обнаруживается, что они образуют «неупорядоченную» пару. Это будет сигналом для вызова программы перемены мест (строка 330). Строка 322 вызывает модифицированную подпрограмму сравнения. Если результатом работы этой программы будет K=0, то строка 323 отправит нас в обход вызова в строке 330 — перемена местами не требуется, и для сравнения будет выбрана следующая пара элементов массивов.

Подпрограмма сравнения (строки 345—369), к радости виртуозов экономии памяти, выполнена почти без комментариев для самостоятельного прочтения апологетами абсолютно плотного кода. Для всех же остальных предлагается такая схема работы подпрограммы (рис. 2).

Оператор в строке 346 опрокидывает флаг в «упорядоченное» положение. В переменные L¤ и M¤ помещаются первые символы сравниваемых элементов массивов. Если (строка 349) хотя бы одна из фамилий не начинается с двоеточия (содержит русские буквы), надо начинать сравнивание по нашему алфавиту. Если же обе фамилии начались с двоеточия (латинские буквы), нам достаточно обычного сравнения в строке 350. Цикл сравнения кириллицы (строки 345—366) отделяет в переменные L¤ и M ¤ очередные символы для сравнения, производящегося в строках 357—360. Указатели IM и IP устанавливаются (см. рис. схемы подпрограммы) при совпадении очередных сравниваемых литер с литерами алфавита в строках 52—56.

Строка 361 отсеивает случай упорядоченного следования пары и, оставив флаг K опущенным, уводит на выход. Строка 362 реагирует на равенство пары сравниваемых литер и отправляет на следующую попытку сравнения следующей пары литер. Строка 363 фиксирует найденное нарушение следования по алфавиту, поднимая флаг K, кото-

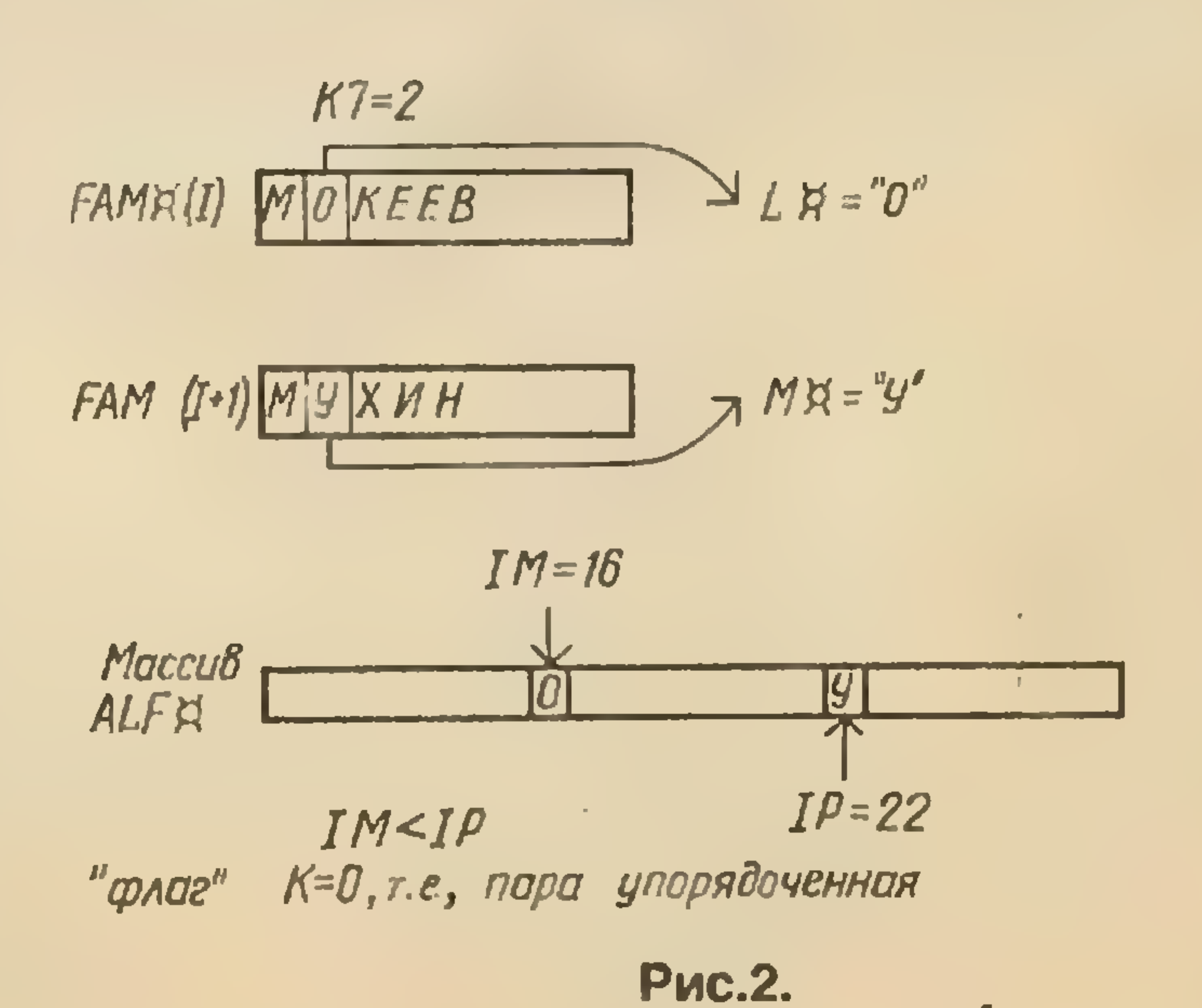



Рис.2.

рым далее воспользуется подпрограмма сортировки в строке 323.

Оглядев в заключение критическим взором проделанную работу, можно найти достаточно мелких и крупных поводов для доработки, для определения новых функций и всевозможной оптимизации. Текст получился хорошо переносимый на разные типы ПЭВМ (мобильный). Применительно к конкретному диалекту языка можно, например, уменьшить длину текста, располагая по несколько операторов в строке (например, короткий цикл в строках 60—64 или определения массивов). При этом, конечно, не надо впадать в другую крайность, жертвуя простотой восприятия текста.

Некоторые диалекты допускают только очень короткие (одна-две буквы) имена переменных. В этом случае можно сократить до минимума количество промежуточных переменных, хотя при дальнейшем развитии программы это может привести к ошибкам. Переменная M при вызове модуля испортит независимое значение переменной M в модуле 1

```
200 FOR M=1 то...
...
250 GOSUB 500                    МОДУЛЬ 1
...
300 NEXT M
            |
            v
500 REM ///
...
600 GOSUB 800                    МОДУЛЬ 2
            |
            v
800 REM ///
...
810 FOR M=2 то...                МОДУЛЬ 3
...
850 NEXT M
```

Существуют гораздо более быстрые (но занимающие больший участок текста) сортировки. Некоторые конструкции можно перепрограммировать более компактно (но и более трудно для прочтения). Однако общий вывод заключается в том, что программа построена с учетом требований структурного программирования, вследствие чего мы имеем реальную возможность прилагать дальнейшие усилия для ее улучшения.

## В ЧИТАЛЬНОМ ЗАЛЕ

1. Ван Тассел Д. Стиль, разработка, эффективность, отладка и испытание программ. — М.: Мир, 1985.

2. Майер Г. Надежность программного обеспечения. — М.: Мир, 1980.

3. Керниган Б. В., Плоджер Р. В. Элементы стиля программирования. — М.: Радио и связь, 1983.




С. М. КОСЕНКОВ

# НОВЫЕ МОДЕЛИ СЕМЕЙСТВА БК

В настоящее время выбор отечественных микрокомпьютеров небогат — это «Микроша», созданный на базе микропроцессорного комплекта КР580, «Электроника БК-0010», использующий микропроцессор К1801. Скоро на прилавках появится набор для сборки радиолюбительской микро-ЭВМ «Радио-86РК». Однако среди этих моделей микро-ЭВМ семейства БК заслуживают особого внимания. И дело здесь не только в том, что это не 8-, а 16-разрядная ЭВМ или что имеется цветная графика, хотя для пользователя все это немаловажно. Главным достоинством этого семейства, очевидно, следует считать ее архитектурную и программную совместимость с большим парком микро- и мини-ЭВМ, выпускаемых промышленностью. Машины семейства БК — младшие братья таких профессиональных компьютеров, как ДВК, УКНЦ, «Электроника-85», СМ-4, СМ-1420, и многих других. Такая совместимость позволяет при соответствующей адаптации использовать многие программы, разработанные для других ЭВМ. Кроме того, в этом случае профессиональные ЭВМ можно использовать в качестве инструментальной машины, при разработке программ для БК.

В настоящее время известны одна модель этого семейства — БК-0010 и ее модификации БК-0010Ш, БК-0010.01. Эти машины уже заняли свое место в народном хозяйстве. Они с успехом используются в сфере образования, производства и, конечно, в быту. Для машин этой модели уже разработан довольно большой фонд программ, включающий инструментальные программы, языковые средства, обучающие системы, игры и др. (рис. 3).

На очереди следующая модель — БК-

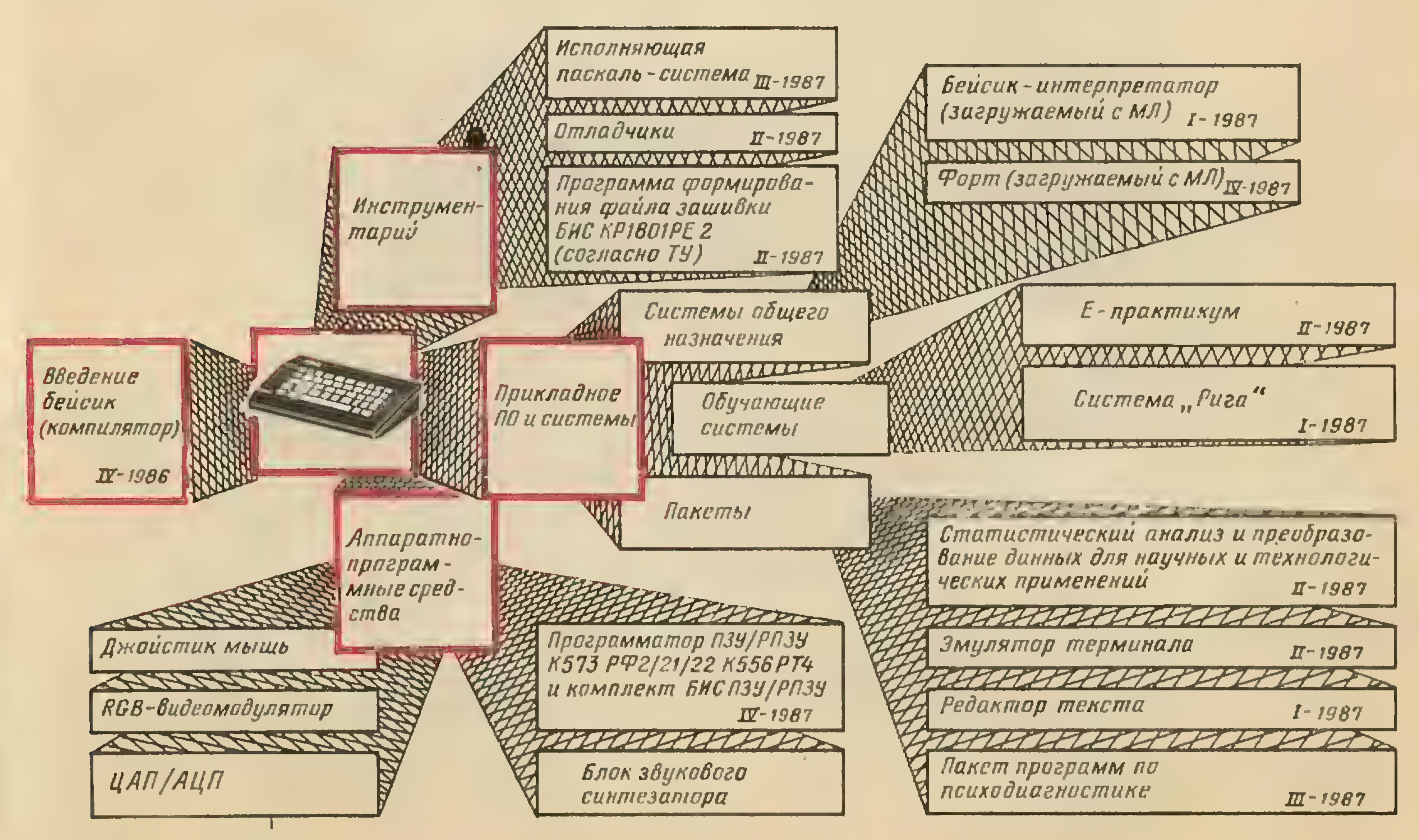



Рис.3.

0011, разработанная и планируемая к выпуску в рамках развития этого семейства микро-ЭВМ. Новая модель отличается от БК-0010 расширенными возможностями графики, большим объемом памяти, широким набором периферийных устройств. Конструктивно (рис. 4) эта модель схожа с БК-0010.01. Используются тот же корпус и нажимная клавиатура. Функциональные возможности БК-0011 расширяются при помощи набора контроллеров (на рис. 7 справа), подключаемых к разъемам основного модуля. С помощью этих контроллеров совместно с микро-ЭВМ можно использовать печатающее устройство, накопитель на гибких магнитных дисках. Контроллер локальной сети позволит объединять машины в сеть типа «кольцо», а контроллер последовательного интерфейса — в сеть типа «звезда».

У БК-0011 значительно больший объем памяти. Пользователь получает постраничный доступ к ОЗУ 128 К байт и ПЗУ 40 К байт, причем объем ПЗУ можно довести до 64 К байт. Расширены возможности графики — две страницы по 16 К байт ОЗУ используются в качестве буферов для графического дисплея. Имеется устройство управления цветовой палитрой, позволяющее создать на экране цветного видеомонитора гамму из 16 цветов.

Наличие накопителя на гибких магнитных дисках и достаточный объем памяти позволили поставить на БК-0011 операционную систему, совместимую с известной зарубежной ОС РТ-11 версии 05.02. Разработан мощный и удобный экранный редактор текстов. В среде операционной системы имеется богатый набор программ, систем программирования ПАСКАЛЬ, ФОРТРАН, СИ, БЕЙСИК, Макро-ассемблер, ПРОЛОГ, ЛИСП, ФОРТ, МОДУЛА-2 и др. Наличие дисковода значительно расширяет области применения БК-0011 и позволяет превратить ее в довольно мощную инструментальную машину. Но и в случае, когда дисковод отсутствует, на новой модели можно эксплуатировать развитую версию БЕЙСИКа, «зашитую» в ПЗУ.

На базе микро-ЭВМ БК-0011 разработан новый комплект учебной вычисли-

*Ряд БК. Конструктивные исполнения*

БК-0010Ш
ИРПС
БК-0011
КПУ
КРМП
КНГМД
КЛС
КМСТД
Корпус БК-0010
Блок клавиатуры ПКН-131(ПКН-111)
Корпус БК-0010
Блок клавиатуры ПКН-111 (ПКН-131)
БК-0010
Блок клавиатуры на ПКН-150
БК-0010М
Блок клавиатуры пленочный
Корпус БК-0010
БК-100
Корпус типа УКНЦ
Блок клавиатуры с полноходовыми кнопками

**Рис.4.**

тельной техники КУВТ-87 (рис. 5). В более раннем комплекте КУВТ-86 использовались модели БК-0010. Они позволяют организовать сеть только по схеме «звезда»; БК-0010 на рабочем месте ученика (РМУ) и центральная машина ДВК-2М на рабочем месте преподавателя (РМП). На новой машине сеть можно строить по схеме «кольцо», и на РМУ и на РМП установлены однотипные БК-0011. В такой сети любая выбранная машина может быть главной (она снабжается печатающим устройством и накопителем на гибких магнитных дисках). Такое решение позволяет значительно повысить надежность всего комплекса в целом.

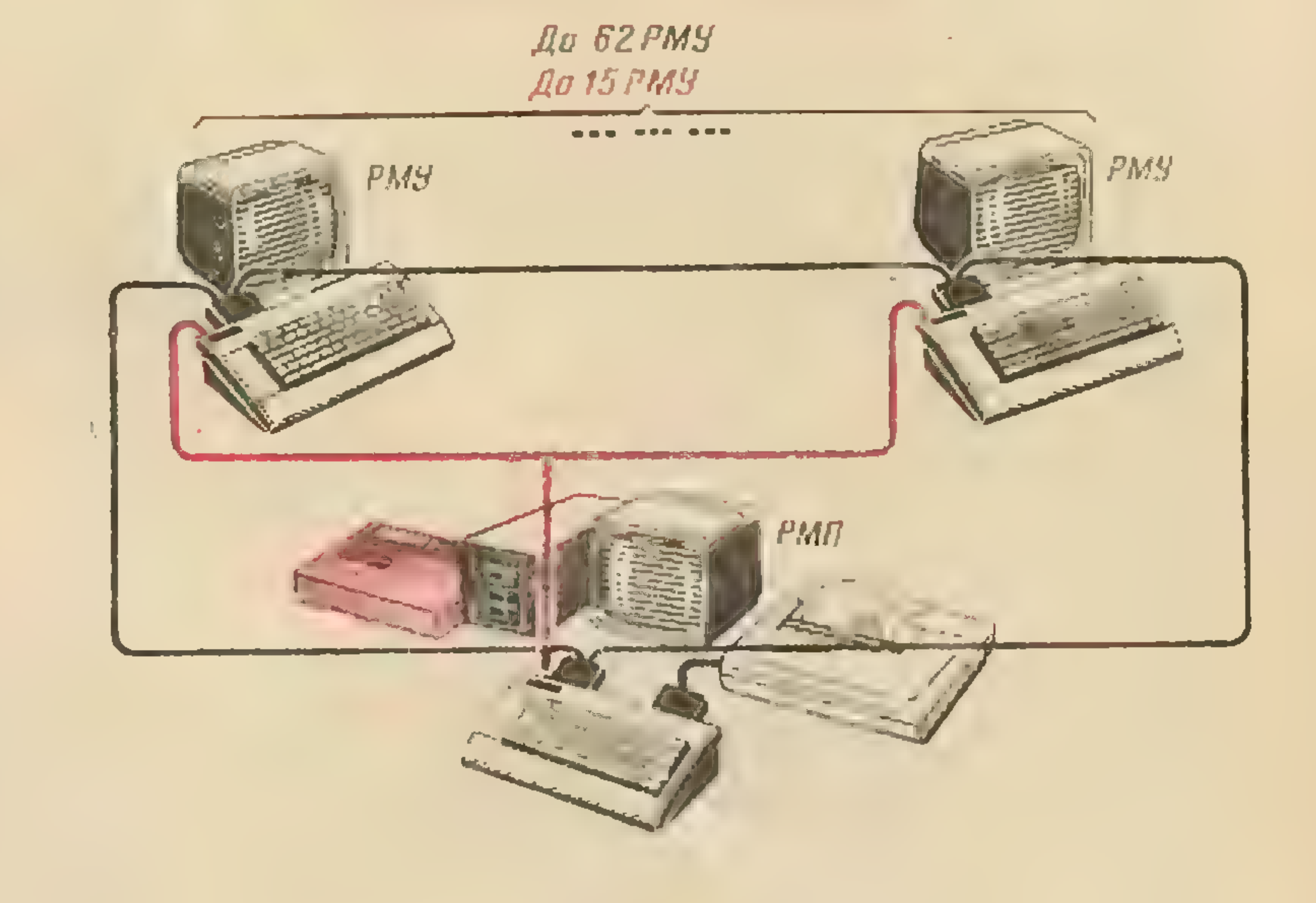



Рис.5.

**КУВТ. Технические характеристики**

| Характеристика | КУВТ-86 БК-ДВ | КУВТ «БК-БК» | КУВТ «БК-0011-УКНЦ» | КУВТ «БК-0011-БК-0011» |
|---|---|---|---|---|
| Тип вычислительной сети | Звезда | Звезда | Кольцо | Кольцо |
| Количество РМУ | 12 | 15 | До 62 | До 62 |
| Тип линии связи | Витая пара | Витая пара | Витая пара | Витая пара |
| Состав РМУ | «Электроника БК-0010», УОИ на базе телевизора «Электроника-Ц 431» или «Электроника-404», блок ИРПС | «Электроника БК-0010», УОИ на базе телевизора «Электроника-Ц 431» или «Электроника-404» | «Электроника БК-0011», УОИ на базе телевизора «Электроника-Ц 431» или «Электроника-404», контроллер локальной сети | «Электроника БК-0011», УОИ на базе телевизора «Электроника-Ц 431» или «Электроника-404», контроллер локальной сети |
| Состав РМП | «Электроника МС-1201.01», дисплей «15ИЭ-00-013», накопитель «Электроника НГМД-6022», устройство печатающее «Электроника УВВ ПЧ 30-004» | «Электроника БК-0010», УОИ на базе телевизора «Электроника-Ц 431» или «Электроника-404», магнитофон «Электроника-302», печатающее устройство | «Электроника УКНЦ» УОИ «Янтарь-8», контроллер локальной сети, накопитель «Электроника НГМД-6022», печатающее устройство Д100 | «Электроника БК-0011», УОИ на базе телевизора «Электроника-Ц 431» или «Электроника-404», контроллер РМП, накопитель «Электроника НГМД-6022», печатающее устройство Д100 |
| Скорость обмена по линиям связи, бод | 9 600 | 1 200 | 57 600 | 57 600 |
| Ориентировочная цена, тыс. руб. | 26 | 16 | 25 (для 16 РМУ) | 23,5 (для 16 РМУ) |

Работа над семейством БК продолжается, и сейчас ведется разработка еще одной модели микро-ЭВМ — БК-0100. Эта машина уже значительно отличается от всех предыдущих моделей. В ней предполагается использовать два различных процессора: один — серии 1801, другой — 580 или 1801. Таким образом, пользователь БК-0100 сможет эксплуатировать программное обеспечение двух различных семейств микропроцессоров. Быстродействие машины возрастет до 0,5 млн. коротких операций в секунду, объем памяти увеличен до 256 К байт. На рис. 7 показано конструктивное исполнение БК-0100.

В следующих выпусках мы подробнее расскажем о модели персональной бытовой микро-ЭВМ БК-0011, о ее конструкции, архитектуре и программном обеспечении.

## Микро-ЭВМ „Электроника БК-0011“

Технические характеристики:
— тип процессора — К 1801 ВМ1;
— разрядность процессора — 16 бит;
— тактовая частота — 3 МГц;
— организация памяти — страничная;
— объем страницы ОЗУ, ПЗУ — 16 К байт;
— объем ОЗУ — 128 К байт (8 страниц), из них экранная память — 16-32 (1-2 страницы);
— объем резидентного ПЗУ — до 64 К байт;
— объем внешнего ПЗУ — 32 К байт (2 страницы).

Формат изображения:
— в алфавитно-цифровом режиме 24 (28) строки текста по 32 символа (черно-белый или цветной монитор) или 64 символа (черно-белый монитор);
— в графическом режиме — 128х256 точек (16 цветов), 256х256 точек (4 цвета), 512х256 точек (черно-белый).

Режимы вывода изображения:
— вывод черно-белого изображения;
— вывод цветного изображения;
— тип НГМД — «Электроника-6022»;
— тип печатающего устройства — D-100 (ПНР);
— тип УОИ — «Электроника-Ц431», «Электроника-404»;
— скорость обмена с магнитофоном — 1200 бод;
— тип кассеты — МК-60;
— скорость обмена по локальной сети — 57600 бод;
— разрядность параллельного программируемого интерфейса — 16 бит;
— тип межмодульного параллельного интерфейса — по ОСТ 11.305.309-80;
— тип клавиатуры — на базе переключателей ПКН 111 (ПКН 131);
— элементная база — БИС серий К 1801, К 565, ИС серий К 561, К 555, К 155;

— наработка на отказ — 5000 ч.

## Основные этапы развития компьютеров типа БК (до 1990 г.)

Основные характеристики:
— объем ОЗУ-32 К байт, из них ОЗУ экрана — 16 К байт;
— объем ПЗУ-24 — 32 К байт;
— тип процессора — К 1801 ВМ 1.

Программное обеспечение:
— драйвер-мониторная система;
— текст — мониторная система;
— ФОКАЛ;
— прикладные программы.

Возможность подключения к внешним устройствам:
— устройство отображения информации на базе телевизионного приемника;
— накопитель — кассетный магнитофон;
— клавиатура — на базе ПКН-150.
Цена — 600 руб.
Год разработки — 1983
Начало серийного производства — 1984

БК-0010 Ш
Основные характеристики:
— объем ОЗУ-32 К байт, из них ОЗУ экрана — 16 К байт;
— тип процессора — К 1801 ВМ 1;
— объем ПЗУ-32 К байт.

Программное обеспечение:
— драйвер-мониторная система БК 0010;
— БЕЙСИК с сетевыми функциями;
— прикладные программы.
Возможность подключения к внешним устройствам:
— УОИ — на базе ТВ-приемника черно-белого или цветного изображения;
ДВК-2МШ — через блок ИРПС;
— клавиатура — на базе ПКН 111.
Цена — 750 руб.
Год разработки — 1986.
Начало серийного производства — 1986.

БК-0011
Основные характеристики:
— объем ОЗУ — 128 К байт (8 страниц 16 Кбайт),
из них ОЗУ экрана — 2 страницы;
— объем ПЗУ — 40 К байт;
— возможность подключения дополнительного внешнего блока ПЗУ — 32 К байт;
— тип процессора — К 1801 БМ1.

Состав микро-ЭВМ:

— устройство информационно-вычислительное;
— блок контроллера НГМД;
— блок контроллера локальной сети;
— блок контроллера рабочего места преподавателя;
— блок контроллера печатающего устройства;
— блок питания.

Программное обеспечение:
— драйвер-мониторная система;
— БЕЙСИК;
— дисковая операционная система ДОС-БК (совместимая с RT-11 V0.5. 0.2);
— пакет прикладных программ;
— клавиатура на базе ПКН-111.
Ориентировочная цена — 1000—1200 руб.
Год разработки — 1986.
Начало серийного производства — 1987.

БК-0100
Основные характеристики:
— объем ОЗУ-128 — 256 К байт;
— объем ПЗУ-40-72 К байт;
— тип процессора — К 1801 ВМ 3, К 580 ВМ 86.

Программное обеспечение:
— драйвер-мониторная система;
— ТЕСТ-мониторная система;
— БЕЙСИК;
— дисковая операционная система RT 11;
— дисковая операционная система MS DOS;
— клавиатура типа УКНЦ.

Состав внешних устройств:
— УОИ — на базе ТВ-приемника (черно-белый или цветной)
— НГМД;
— печатающее устройство;
— манипуляторы (типа «джойстик», «мышь»).
Ориентировочная цена — 1400 руб.
Год разработки — 1988.
Начало серийного производства — 1989.

марки
**ТИПЫ**
характеристики



## УВАЖАЕМЫЕ ЧИТАТЕЛИ!

**Заканчивается первый год выхода в свет подписных научно-популярных брошюр серии „Вычислительная техника и ее применение". С начала нового года издания (с января 1989 г.) можно изменить формат изданий, подбор тем, художественное оформление, стиль изложения и др. Ждем ваших замечаний, пожеланий и рекомендаций.**

периодические издания

## ЭВМ

## Микропроцессоры

## Электроника

Продолжение. Начало см. выпуски 2 и 4

* АВТОМАТИКА И ТЕЛЕМЕХАНИКА. Орган Академии наук СССР. Журнал основан в 1936 г. Выходит в свет 12 раз в год. Издательство «Наука».

* АВТОМАТИКА, ТЕЛЕМЕХАНИКА И СВЯЗЬ. Ежемесячный массовый производственно-технический журнал. Орган Министерства путей сообщения СССР. Основан в 1923 г. Москва, издательство «Транспорт».

* БЮЛЛЕТЕНЬ ИНОСТРАННОЙ НАУЧНО-ТЕХНИЧЕСКОЙ ИНФОРМАЦИИ ТАСС. Новости науки и техники по сообщениям иностранной периодической печати, но без оценки достоверности сообщений.

* ВЕСТНИК МОСКОВСКОГО УНИВЕРСИТЕТА. Научный журнал. Серия 15 — Вычислительная математика и кибернетика. Основан в 1946 г. Выходит 4 раза в год. Издательство Московского университета.

* ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ. Сборник трудов Сибирского отделения АН СССР. Издания выходят с 1961 г.

* ЭКСПРЕСС-ИНФОРМАЦИЯ. ЗАРУБЕЖНАЯ ТЕХНИКА СВЯЗИ. Серия «Телефония, телеграфия, передача данных».

* ИЗВЕСТИЯ АКАДЕМИИ НАУК СССР. Серия «Техническая кибернетика». Журнал основан в 1963 г. Выходит 6 раз в год. Издательство «Наука».

* ПРОБЛЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ. Журнал Академии наук СССР. Основан в 1965 г. Выходит 4 раза в год. Издательство «Наука».

* ПРОБЛЕМЫ ТЕОРИИ И ПРАКТИКИ УПРАВЛЕНИЯ. Издание Международного научно-исследовательского института проблем управления и научно-производственного объединения «Центропрограммсистем».

* ПРОГРАММНЫЕ ПРОДУКТЫ И СИСТЕМЫ. Приложение к журналу «Проблемы теории и практики управления». Периодичность — 4 раза в неделю. Издается с 1988 г. Освещает практические вопросы использования программ в самых различных областях деятельности человека.

* РАДИОТЕХНИКА И ЭЛЕКТРОНИКА. Журнал Академии наук СССР. Основан в 1965 г. Выходит 12 раз в год. Издательство «Наука».

* РЕФЕРАТИВНЫЕ ЖУРНАЛЫ. Орган Государственного комитета СССР по науке и технике и Академии наук СССР. Издания Всесоюзного института научной и технической информации. Издаются (каждое название) 12 раз в год. Среди них:

* АВТОМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
* ВОЛОКОННО-ОПТИЧЕСКИЕ СИСТЕМЫ
* ВЫЧИСЛИТЕЛЬНЫЕ НАУКИ
* ИНФОРМАТИКА
* МЕТРОЛОГИЯ И ИЗМЕРИТЕЛЬНАЯ ТЕХНИКА
* ПРОМЫШЛЕННЫЕ РОБОТЫ И МАНИПУЛЯТОРЫ
* РАДИОТЕХНИКА
* ТЕХНИЧЕСКАЯ КИБЕРНЕТИКА
* ЭЛЕКТРОНИКА
* ЭЛЕКТРОСВЯЗЬ
* БИОНИКА, БИОКИБЕРНЕТИКА, БИОИНЖЕНЕРИЯ

Отдельные статьи смотри в журналах:

* В МИРЕ НАУКИ. SCIENTIFIC AMERICAN. Иллюстрированный журнал. Перевод с английского. Выходит 12 раз в год. Рубрики: Математические игры, Занимательный компьютер и др. Издается с 1983 г. Издательство «Мир».

* МИР НАУКИ. Орган Всемирной федерации научных работников. Выходит с 1957 г. Издается ежеквартально на русском, английском, немецком языках и на эсперанто. Издательство ВЦСПС, Профиздат.

* ВОПРОСЫ ФИЛОСОФИИ. Научно-теоретический журнал. Издается с 1947 г. Выходит ежемесячно. Орган Академии наук СССР. Институт философии. Издательство «Правда».

* КОММУНИСТ. Теоретический и политический журнал Центрального комитета КПСС. Выходит 1 раз в 20 дней. Издается с 1924 г.

**Семейство** отечественных ЭВМ. — М.: Знание,
С 30 1988. — 48 с. — (Новое в жизни, науке, технике. Сер. „Вычислительная техника и ее применение“; № 7).
15 к.

В брошюре приведены сведения о разработке, производстве и перспективе развития семейства микро-ЭВМ, широко распространенных в настоящее время под названием бытовой компьютер — БК-0010 и БК-0011. Описаны особенности эксплуатации этих популярных моделей микро-ЭВМ, их характеристики, программное обеспечение.

Рассчитана на студентов, преподавателей, лекторов, инженеров и производственников, всех читателей, интересующихся вопросами программирования и эксплуатации средств вычислительной техники.

**2404000000** **ББК 32.97**

# ТЕМА СЛЕДУЮЩЕГО ВЫПУСКА

## Универсальный язык — возможно ли это?

Научно-популярное издание

СЕМЕЙСТВО ОТЕЧЕСТВЕННЫХ ЭВМ

Гл. отраслевой редактор Л. А. Ерлыкин
Редактор Б. М. Васильев
Мл. редактор Н. А. Васильева
Художники В. Н. Конюхов и К. Н. Мошкин
Худож. редактор М. А. Гусева
Техн. редактор А. М. Красавина
Корректор В. В. Каночкина
ИБ № 9227

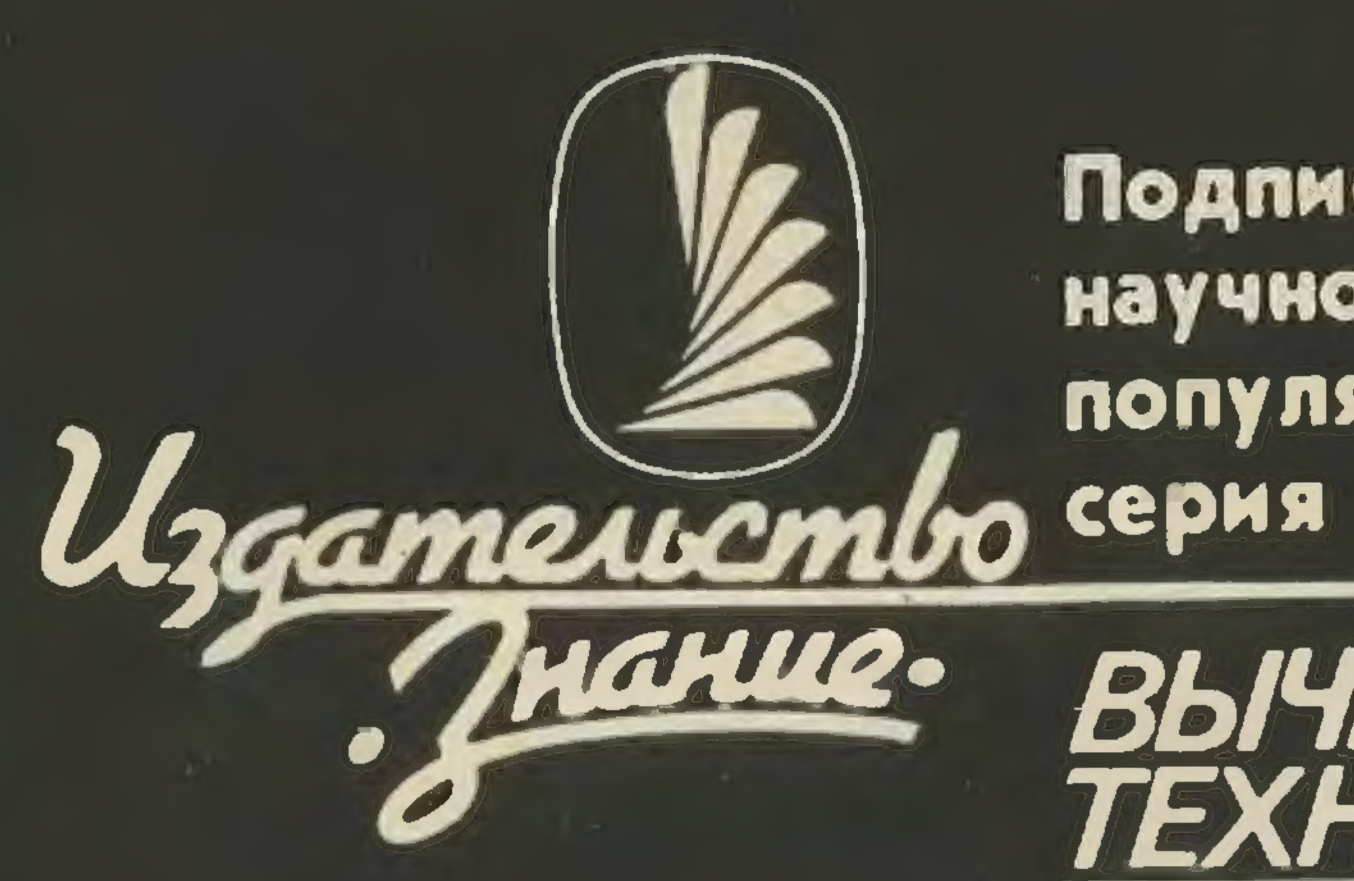
Сдано в набор 23.03.88. Подписано к печати 19.05.88. Т — 01462. Формат бумаги 70×100 1/16. Бумага офсетн. № 2. Гарнитура гельветика. Печать офсетная. Усл. печ. л. 3,90. Усл. кр.-отт 8,45. Уч.-изд. л. 4,10. Тираж 65 820 экз. Заказ 1984. Цена 15 коп. Издательство «Знание». 101835, ГСП, Москва, Центр, проезд Серова, д. 4. Индекс заказа 884707.
Ордена Трудового Красного Знамени Калининский полиграфический комбинат Союзполиграфпрома при Государственном комитете СССР по делам издательств, полиграфии и книжной торговли. 170024, г. Калинин, пр. Ленина, 5.

Цена 15 коп.

Индекс 70195

*Адрес подписчика:*

Лен 27-43




Подписная научно-популярная серия

# ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ЕЁ ПРИМЕНЕНИЕ

*Дорогой читатель!*

*Брошюры этой серии в розничную продажу не поступают, поэтому своевременно оформляйте подписку. Подписка на брошюры издательства «Знание» ежеквартальная, принимается в любом отделении «Союзпечати».*

*Наш адрес: СССР, Москва, Центр, проезд Серова, 4*